

PROGRAMMABLE DC TO DC CONVERTER
PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh :

Ela sari	NIRM	0031639
Fahmi Hidayat	NIRM	0031640

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG**
TAHUN 2019

LEMBAR PENGESAHAN

JUDUL PROYEK AKHIR PROGRAMMABLE DC TO DC CONVERTER

Disusun Oleh :

Ela sari 0031639

Fahmi Hidayat 0031640

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing I

I Made Andik Setiawan, Ph.D

Pembimbing II

Eko Sulistyo, M.T

Pengaji I

Yudhi, M.T.

Pengaji II

Zanu Saputra, M.T.

Pengaji III

Nofriyani, M.T.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan dibawah ini :

Nama Mahasiswa I	:	Ela Sari	NIRM 0031639
Nama Mashasiswa II	:	Fahmi Hidayat	NIRM 0031640

Dengan Judul : PROGRAMMABLE DC TO DC CONVERTER

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, Agustus 2019

Nama Mahasiswa

1. Ela Sari

Tanda Tangan



.....

2. Fahmi Hidayat



.....

ABSTRAK

Photovoltaic menghasilkan tegangan berubah-ubah tergantung dengan intensitas cahaya yang diterima. Tegangan yang beubah-ubah atau tidak stabil ini akan berpengaruh terhadap beban. Dari permasalahan tersebut Programmable DC to DC Converter ini dirancang untuk menghasilkan tegangan searah (Direct Current) variabel yang dapat menstabilkan tegangan dengan tegangan output yang bisa diatur dengan program arduino sesuai dengan kebutuhan beban. Converter yang dirancang dapat menurunkan dan menaikan tegangan (buck-boost converter). Metode yang digunakan dalam converter ini adalah metode pensaklaran atau switching. Metode switching ini akan mendapatkan tegangan dengan tingkat efisiensi tinggi dibandingkan dengan metode linear. Pada sistem ini pengontrolan menggunakan kontrol Proportional Integral Derivative (PID) yang diterapkan pada DC to DC converter. Penerapan dari Proportional Integral Derivative (PID) menghasilkan respon yang cepat dan sedikit overshoot pada keluaran DC to DC converter, tegangan keluar lebih presisi, meminimalisir presentase error. Dari data pengujian serta analisa menunjukan dengan nilai konstanta proportional bernilai 1, konstanta integral bernilai 0,1 dan konstanta derivative bernilai 0,1 maka respon yang dihasilkan cepat dan sedikit overshoot yang dihasilkan.

Kata kunci: DC to DC Converte, Pensaklaran, PID

ABSTRACT

Photovoltaic produce variable voltages based on the intensity of the light received. The variable voltages or unstable will effect to load. The problem above the Programmable DC to DC Converter is designed to produce variable dc voltages with output voltage that can be adjusted with the Arduino program by to the needs. Converter is designed can up and down the voltage. The method used in this converter is the switching. With this method will get a voltage with high level of efficiency than with linear method. This final project carried out a study of control Proportional integral Derivative (PID) applied to DC Converter. (Application of Proportional integral Derivative (PID) results in a faster response, less overshoot on the converter output voltage, voltage is more precise, and minimize the percentage of errors. From the test data and analysis shows the proportional constant value is 1, Integral constant value 0,1 and Derivative constant value is 0,1, then the resulting response is fast and the resulting overshoot is small.

Keywords: DC to DC Converter, Switching, PID

KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadirat Allah SWT, atas berkat dan rahmatnya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini.

Laporan Tugas Akhir ini disusun sebagai salah satu persyaratan dan kewajiban mahasiswa untuk menyelesaikan kurikulum program pendidikan Diploma III di Politeknik Manufaktur Negeri Bangka Belitung.

Penulis mencoba untuk menerapkan ilmu pengetahuan yang telah dipelajari selama 3 tahun mengecap pendidikan di Politeknik Manufaktur Negeri Bangka Belitung dan pengalaman penulis selama program Praktek Kerja Lapangan pada pembuatan alat dan makalah tugas akhir ini.

Makalah ini dapat diselesaikan dengan adanya usaha dan kerja tim yang baik serta bantuan, saran-saran dan informasi dari berbagai pihak. Pada kesempatan ini penulis ingin mengucapkan terimakasih kepada pihak-pihak tersebut, antara lain.

1. Bapak I Made Andik S, M.Eng, Ph.D selaku pembimbing I yang telah meluangkan waktu, teganga dan pikiran didalam mengarahkan pengarahan dalam penulisan Laporan Tugas Akhir ini.
2. Bapak Eko Sulistyo, M.T selaku pembimbing II yang telah memberikan saran-saran dan solusi dari masalah-maslah yang dihadapi selama proses perencanaan dan pembuatan alat serta penyusunan laporan.
3. Bapak Surojo, M.T selaku Wali Kelas III Elektronika B Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung
4. Seluruh staf pengajar di Politeknik Manufaktur Negeri Bangka Belitung
5. Bapak Sugeng Ariyono, M.Eng, Ph.D selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung
6. Rekan-Rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah banyak membantu selama penggeraan Tugas Akhir

7. Keluarga besar yang selalu senantiasa memberikan kasih sayang, doa, serta dukungan moril maupun materil.
8. Pihak-pihak lain yang telah memberikan bantuan baik secara langsung maupun tidak langsung dalam pembuatan Tugas Akhir ini yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa penulisan Laporan ini masih jauh dari sempurna dikarenakan penulis adalah manusia yang tidak luput dari salah. Oleh karena itu diharapkan segala kritik dan saran agar kedepannya penulis dapat mengembangkan dan memperbaiki penulisan selanjutnya.

Akhir kata mengucapkan terima kasih. Semoga Laporan Tugas Akhir ini dapat berguna untuk menambah wawasan bagi penulis dan pembaca.

Sungailiat,28 Agustus 2019

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	ii
PERNYATAAN BUKAN PLAGIAT	iii
ABSTRAK	iv
<i>ABSTRACT</i>	v
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah.....	2
1.4 Tujuan Proyek Akhir	2
BAB II DASAR TEORI	3
2.1 <i>DC to DC Converter</i>	3
2.2 Jenis-jenis <i>DC to DC Converter</i>	3
2.2.1 <i>Buck Converter</i>	3
2.2.2 <i>Boost Converter</i>	6
2.2.3 <i>Buck-Boost Converter</i>	8
2.3 <i>Pulse Width Modulation</i>	10
2.4 Kontrol PID (<i>Proportional–Integral–Derivative</i>).....	11
2.4.1. Kontrol P (<i>Proportional</i>)	11
2.4.2. Kontrol I (<i>Integral</i>)	12
2.4.3. Kontrol D (<i>Derivative</i>).....	12
BAB III METODE PELAKSANAAN	14
3.1 Komponen yang digunakan.....	16
BAB IV PEMBAHASAN.....	17

4.1	<i>DC to DC Converter</i>	17
4.2	Sensor Tegangan	20
4.3	<i>Liquid Crystal Display (LCD)</i>	23
4.4	<i>Keypad</i>	28
4.5	Pembuatan <i>Box</i>	32
4.6	Rangkaian Kontrol	32
4.7	Kontrol <i>Proportional (P)</i>	33
4.8	Kontrol <i>Proportional Integral (P-I)</i>	35
4.9	Kontrol <i>Proportional Integral Derivative</i>	38
BAB V KESIMPULAN		49
5.1	Kesimpulan	49
5.2	Saran	49
DAFTAR PUSTAKA		50

DAFTAR TABEL

Tabel 4. 1 Pengujian Sensor Tegangan	22
Tabel 4. 2 Percobaan Metode <i>Trial and Error</i> Konstanta <i>Prortional</i> (K_p)	34
Tabel 4. 3 Percobaan Metode <i>Trial and Error</i> Konstanta <i>Prortional</i> (K_p) dan Konstanta <i>Integral</i> (K_i)	36
Tabel 4. 4 Nilai K_p dan K_i tetap, sedangkan nilai K_d berubah-ubah	39
Tabel 4. 5 Nilai K_p dan nilai K_d tetap, sedangkan nilai K_i berubah-ubah	41
Tabel 4. 6 Nilai K_p yang tetap sedangkan nilai K_i dan nilai K_d berubah-ubah.....	42
Tabel 4. 7 Nilai K_p berubah-ubah sedangkan nilai K_i dan K_d tetap	44
Tabel 4. 8 Nilai K_p dan K_d berubah-ubah sedangkan nilai K_i tetap	46

DAFTAR GAMBAR

Gambar 2. 1 Rangkaian Topologi <i>Buck Converter</i>	4
Gambar 2. 2 Rangkaian <i>Buck Converter State On</i>	5
Gambar 2. 3 Rangkaian <i>Buck converter State Off</i>	5
Gambar 2. 4 Grafik Keluaran dari <i>Buck Converter</i>	5
Gambar 2. 5 Rangkaian Dasar dari <i>Boost Converter</i>	6
Gambar 2. 6 Rangkaian <i>Boost Converter</i> dalam Kondisi <i>State On</i>	7
Gambar 2. 7 Rangkaian <i>Boost Converter</i> dalam Kondisi <i>State Off</i>	7
Gambar 2. 8 Grafik Keluaran dari <i>Boost Converter</i>	8
Gambar 2. 9 Rangkaian <i>Buck-Boost Converter</i>	8
Gambar 2. 10 Rangkaian <i>Buck-Boost</i> Saklar Tertutup ..	9
Gambar 2. 11 Rangkaian <i>Buck-Boost Converter</i> Saklar Terbuka.....	9
Gambar 2. 12 Gelombang <i>Pulse Width Modulation</i>	10
Gambar 2. 13 Blok Diagram Kontrol <i>Proportional Integral Derivative (PID)</i> ..	11
Gambar 3. 1 Blok diagram <i>Programmable DC to DC Converter</i>	15
Gambar 4. 1 Rangkaian <i>Schematic Buck-Boost Converter XLSEMI 6009</i>	17
Gambar 4. 2 <i>Converter XLSEMI 6009</i>	18
Gambar 4. 3 Grafik <i>Buck converter</i>	19
Gambar 4. 4 Grafik <i>Boost Converter</i>	19
Gambar 4. 5 Grafik Pengujian Pin Enable.....	20
Gambar 4. 6 Rancangan Sensor Tegangan	21
Gambar 4. 7 Rangkaian <i>Liquid Crystal Display (LCD)</i>	23
Gambar 4. 8 Tampilan Pengujian <i>Liquid Crystal Display (LCD)</i>	24
Gambar 4. 9 <i>Hardware Liquid Crystal Display (LCD)</i>	26
Gambar 4. 10 Tampilan dengan Menu <i>Default</i>	27
Gambar 4. 11 Tampilan Menu <i>Parallax</i>	27
Gambar 4. 12 Tampilan Menu <i>Matlab</i>	27
Gambar 4. 13 Tampilan Menu <i>Set Parameter</i>	28
Gambar 4. 14 Perancangan dari <i>Keypad</i>	28
Gambar 4. 15 Menu-menu yang disediakan	29

Gambar 4. 16 Hasil dan desain <i>Box</i>	32
Gambar 4. 17 Rangkaian Kontrol <i>Programmable DC to DC Converter</i>	33
Gambar 4. 18 Grafik Perbandingan nilai K_p	35
Gambar 4. 19 Grafik Perbandingan nilai K_p dan K_i	37
Gambar 4. 20 Grafik dengan nilai $K_p=1$; $K_i=0,1$; $K_d=0,1$	39
Gambar 4. 21 Grafik dengan nilai $K_p=1$; $K_i=0,1$; $K_d=0,3$	39
Gambar 4. 22 Grafik dengan nilai $K_p=1$; $K_i=0,1$; $K_d=0,5$	40
Gambar 4. 23 Grafik dengan nilai $K_p=1$; $K_i=0,1$; $K_d=0,6$	40
Gambar 4. 24 Grafik dengan nilai $K_p=1$; $K_i=0,1$; $K_d=0,8$	40
Gambar 4. 25 Grafik dengan nilai $K_p=1$; $K_i=0,1$; $K_d=0,1$	41
Gambar 4. 26 Grafik dengan nilai $K_p=1$; $K_i=0,5$; $K_d=0,1$;.....	41
Gambar 4. 27 Grafik dengan nilai $K_p=1$; $K_i=1$; $K_d=0,1$;.....	42
Gambar 4. 28 Grafik dengan nilai $K_p=1$; $K_i=0,1$; $K_d=1$	43
Gambar 4. 29 Grafik dengan nilai $K_p=1$; $K_i=0,3$; $K_d=3$	43
Gambar 4. 30 Grafik dengan nilai $K_p=1$; $K_i=0,7$; $K_d=7$	43
Gambar 4. 31 Grafik dengan nilai $K_p=1$; $K_i=1$; $K_d=10$	44
Gambar 4. 32 Grafik dengan nilai $K_p=0,4$; $K_i=0,1$; $K_d=0,1$;.....	45
Gambar 4. 33 Grafik dengan nilai $K_p=0,8$; $K_i=0,1$; $K_d=0,1$	45
Gambar 4. 34 Grafik dengan nilai $K_p=1$; $K_i=0,1$; $K_d=0,1$;.....	45
Gambar 4. 35 Grafik dengan nilai $K_p=0,1$; $K_i=0,7$; $K_d=0,1$	46
Gambar 4. 36 Grafik dengan nilai $K_p=0,4$; $K_i=0,7$; $K_d=0,4$	47
Gambar 4. 37 Grafik dengan nilai $K_p=0,8$; $K_i=0,7$; $K_d=0,8$	47
Gambar 4. 38 Grafik dengan nilai $K_p=1$; $K_i=0,7$; $K_d=1$	47

DAFTAR LAMPIRAN

Lampiran 1 : Daftar Riwayat Hidup (Perorangan)

Lampiran 2 : *Schematic* Rangkaian Kontrol Keseluruhan

Lampiran 3 : *Data Sheet* DC to DC Converter XL SEMI 6009

Lampiran 4 : *Data Sheet* Sensor Arus INA219

Lampiran 5 : Program Arduino UNO

BAB I

PENDAHULUAN

1.1 Latar Belakang

DC to DC Converter adalah peralatan yang terdiri dari piranti dan komponen elektronik yang digunakan untuk merubah besaran tegangan searah atau DC ke besaran tegangan DC lainnya. Penerapan *DC to DC Converter* biasanya digunakan untuk menghasilkan tegangan searah DC (Direct Current) dengan besaran sesuai kebutuhan beban. Aplikasi *DC to DC Converter* ini sangat dibutuhkan pada peralatan sumber energi terbarukan seperti *Photovoltaic*, *accu*, baterai dan lain-lain.

Photovoltaic menghasilkan tegangan *output* berdasarkan intensitas cahaya yang diterima dan biasanya memanfaatkan energi surya sebagai sumbernya. Intensitas cahaya matahari setiap waktu berubah-ubah yang dipengaruhi beberapa faktor seperti hujan, cahaya matahari tertutup awan maupun kondisi lingkungan, sehingga tegangan yang keluar tidak stabil atau berbeda-beda [1]. Tegangan yang dihasilkan oleh *Photovoltaic* pada pagi hari akan lebih kecil dibandingkan dengan tegangan yang dihasilkan ketika siang hari dikarenakan perbedaan intensitas. Begitu juga dengan baterai dan aki seiring dengan lamanya penggunaan maka daya pada aki dan baterai akan semakin berkurang sehingga tegangan yang dibutuhkan tidak lagi terpenuhi. Penerapan *DC to DC Converter* pada sumber sangat diperlukan untuk menjaga tegangan *output* tetap stabil dan sesuai dengan kebutuhan. Sehingga dari permasalahan tersebut diperlukanlah *Programmable DC to DC Converter*.

1.2 Rumusan Masalah

Berdasarkan pembahasan pada latar belakang rumusan masalah pada proyek akhir ini sebagai berikut :

1. Bagaimana cara membuat *DC to DC Converter* dengan tegangan *outputnya* dapat diatur sesuai dengan kebutuhan beban.

2. Bagaimana cara membuat suatu alat yang dapat menstabilkan tegangan yang berubah-ubah.

1.3 Batasan Masalah

1. Tidak dapat memberikan tegangan yang melebihi tegangan keluaran dari *converter* yaitu 28 volt
2. Tegangan *set point* yang diberikan tidak dapat berupa angka desimal.

1.4 Tujuan Proyek Akhir

Dari rumusan masalah diatas, maka tujuan dari Proyek Akhir ini yaitu.

1. Tegangan referensi akan diatur dengan pemrograman sehingga menghasilkan tegangan *output* yang sama dengan tegangan *input*.
2. Menstabilkan besaran tegangan sumber yang berubah-ubah.

BAB II

DASAR TEORI

2.1 DC to DC Converter

Penggunaan tegangan searah (DC) pada sistem tenaga sangat dibutuhkan, maka dari itu peralatan yang menghasilkan tegangan ini masih terus dikembangkan. Salah satu aplikasi dari pengembangan tersebut adalah *DC to DC Converter* atau sering disebut juga *DC Chopper*. *DC to DC Converter* adalah suatu alat yang penyedia daya tegangan searah (DC) yang dihasilkan melalui konversi tegangan DC masukan kebentuk tegangan DC keluaran yang lebih rendah atau tinggi [1]. Pada suatu *DC to DC Converter* tegangan *output* dikendalikan oleh *duty cycle* dengan waktu pensaklaran yaitu timer on (Ton) dan timer off (Toff). Pengendalian dilakukan dengan memberikan sinyal ke *switch*. Untuk mengatur pengendalian tersebut biasanya menggunakan *Pulse Width Modulation* (PWM).

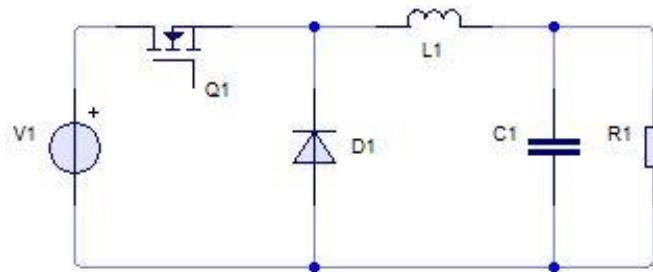
2.2 Jenis-jenis DC to DC Converter

DC to DC Converter yang berfungsi untuk mengkonversikan tegangan masukan searah (DC) konstan menjadi tegangan keluaran searah (DC) lainnya dengan nilai tegangan yang lebih kecil atau lebih rendah. Pada umumnya *DC to DC Converter* dibagi menjadi 3 jenis, yaitu :

2.2.1 Buck Converter

Converter jenis *buck* merupakan jenis konverter yang banyak digunakan dalam industri catu-daya. *Buck converter* atau penurun tegangan adalah salah satu jenis dari *DC to DC Converter* yang mana tegangan *output* selalu lebih rendah dari tegangan *input* [2]. Topologi dari *buck converter* dapat dilihat dari Gambar 2.1. Ada dua mode normal dari *buck converter* tergantung pada arus yang melalui induktor. Mode operasi utama adalah terkarakteristik oleh arus secara terus menerus. Di mode yang lain arus melalui induktor akan mencapai nol dan menunggu hingga sampai sumber listrik mengalir lebih banyak daya ke rangkaian [3]. Untuk membuat

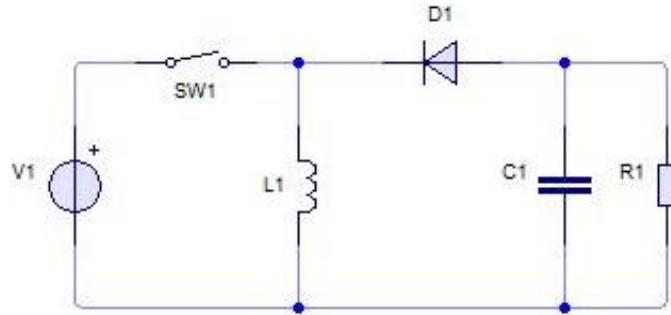
tegangan yang lebih rendah daripada tegangan masukannya, harus menggunakan komponen *switching* untuk mengatur *Pulse Width Modulation* (PWM). Contoh komponen *switching* tersebut adalah thyristor, MOSFET, IGBT dan lainnya. Berikut adalah rangkaian topologi dari *buck converter* yang dapat dilihat pada Gambar 2.1 [4]



Gambar 2. 1 Rangkaian Topologi *Buck Converter* [4]

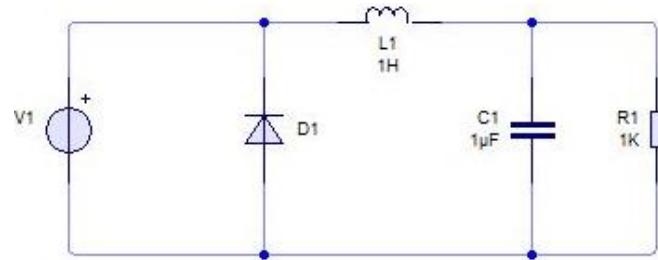
Rangkaian kontrol (*Driver Circuit*) digunakan untuk mengendalikan MOSFET, sehingga MOSFET mengetahui kapan harus membuka dan menutup. Induktor digunakan untuk menyimpan energi dalam bentuk arus energi tersebut disimpan ketika MOSFET dalam kondisi on dan dilepaskan ketika MOSFET dalam kondisi off. Dan dioda digunakan untuk mengalirkan arus yang dihasilkan oleh MOSFET.

Prinsip kerja dari *buck converter* ini terdiri dari 2 *state* yaitu *state on* dan *state off*. Untuk *state on* arus yang mengalir dari sumber menuju ke induktor sehingga induktor menyimpan arus. Setelah kapasitas induktor telah terpenuhi, arus akan disaring (*filter*) oleh kapasitor dan kemudian dialirkan menuju beban dan selanjutnya kembali lagi kesumber. Pada Gambar 2.2 merupakan rangkaian *buck converter* ketika *state on*



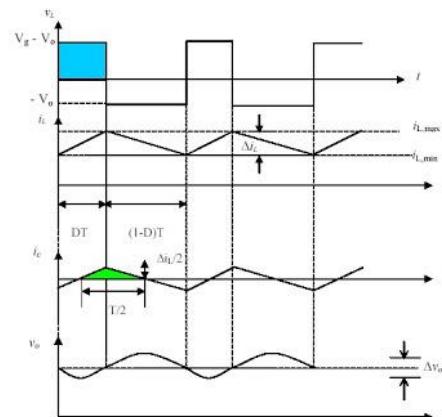
Gambar 2. 2 Rangkaian Buck Converter State On

Sedangkan untuk *state off* arus yang disimpan diinduktor akan dikeluarkan menuju beban lalu menuju dioda dan kembali lagi ke induktor. Gambar 2.3 merupakan rangkaian buck converter ketika state off.



Gambar 2. 3 Rangkaian Buck converter state Off

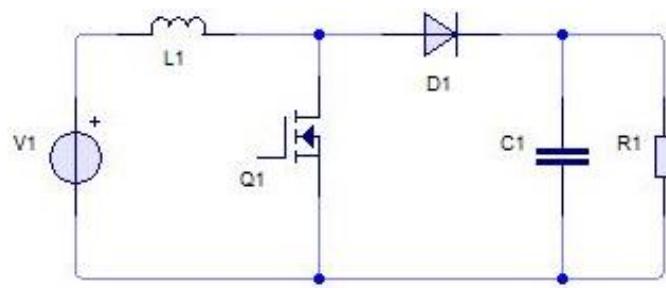
Dari 2 kondisi diatas maka grafik keluaran dari *buck converter* ini dapat dilihat dari Gambar 2.4 sebagai berikut.



Gambar 2. 4 Grafik Keluaran dari Buck Converter

2.2.2 Boost Converter

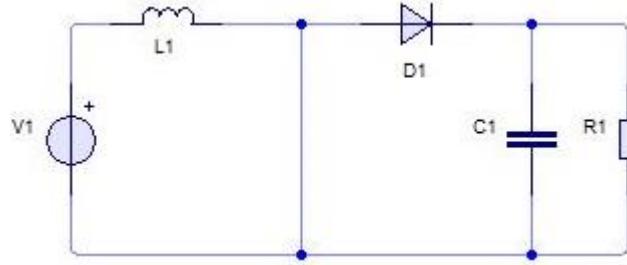
Tegangan *output* dari *boost converter* selalu lebih besar dari tegangan *input* [5]. Sama halnya dengan *buck converter*, Untuk mendapatkan tegangan yang lebih rendah daripada msukannya, harus menggunakan komponen *switching* untuk mengatur *duty cycle*. Komponen *switching* tersebut berupa thyristor, MOSFET, IGBT dan lainnya. Pada Gambar 2.5 merupakan rangkaian dasar dari *boost converter*.



Gambar 2. 5 Rangkaian Dasar dari *Boost Converter* [6]

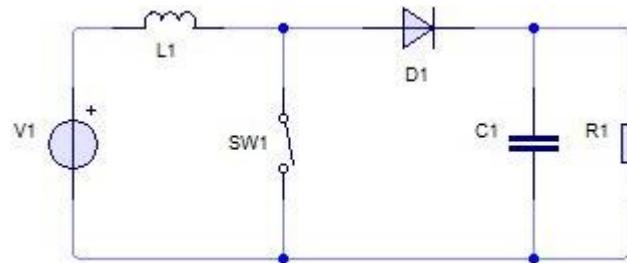
Seperti halnya, Rangkaian *boost* juga mempunyai rangkaian kontrol (*Driver Circuit*) digunakan untuk mengendalikan MOSFET, sehingga MOSFET mengetahui kapan harus membuka dan menutup. Induktor digunakan untuk menyimpan energi dalam bentuk arus energi tersebut disimpan ketika MOSFET dalam kondisi on dan dilepaskan ketika MOSFET dalam kondisi off. Dan dioda digunakan untuk mengalirkan arus yang dihasilkan oleh MOSFET.

Prinsip kerja dari rangkaian *boost converter* memiliki 2 *state*, yaitu *state on* dan *state off*. Untuk *state on* arus akan mengalir searah jarum jam dari sumber menuju ke induktor, induktor akan menyimpan arus. Polaritas induktor pada sisi kiri lebih positif dibandingkan sisi kanannya. Pada Gambar 2.6 merupakan rangkaian *boost converter* dalam oondisi *state On*.



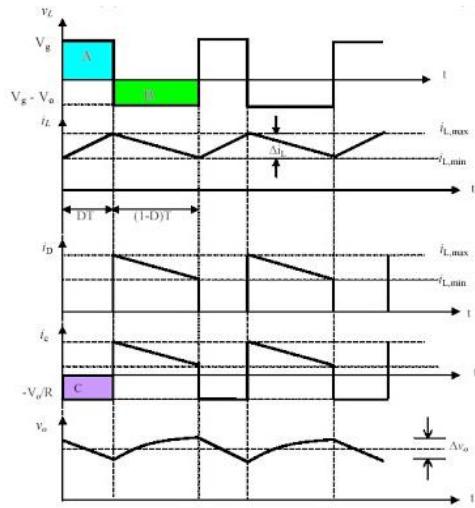
Gambar 2. 6 Rangkaian *Boost Converter* dalam Kondisi *State On*

Sedangkan *State off* arus yang disimpan di induktor akan berkurang karena impedansi yang lebih tinggi. Berkurangnya arus pada induktor akan menyebabkan induktor tersebut dengan melawannya dengan membalik polaritasnya (lebih negatif pada sisi kiri). Sehingga arus yang mengalir pada dioda dan pada beban adalah penjumlahan antara arus sumber dan arus yang ada pada induktor (seri). Disaat yang bersamaan kapasitor akan melakukan penyimpanan energi dalam bentuk tegangan. Itu lah penyebab *boost converter* dapat mengeluarkan tegangan keluar lebih besar dari tegangan masukan. Berikut adalah rangkaian dari *boost converter* dalam kondisi *state off* dapat dilihat pada Gambar 2.7



Gambar 2. 7 Rangkaian *Boost Converter* dalam Kondisi *State Off*

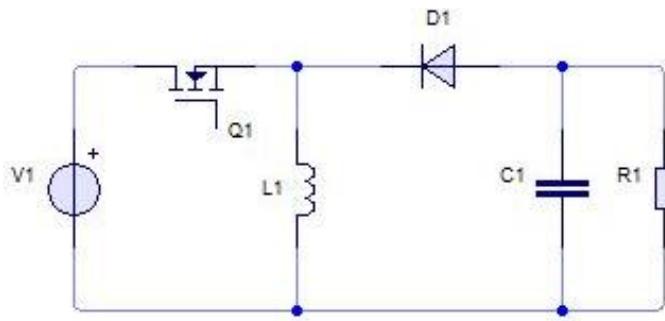
Dari rangkaian *boost converter* diatas didapatkan hasil grafik keluaran berupa gelombang yang ditunjukan pada Gambar 2.8 sebagai berikut.



Gambar 2. 8 Grafik Keluaran dari *Boost Converter*

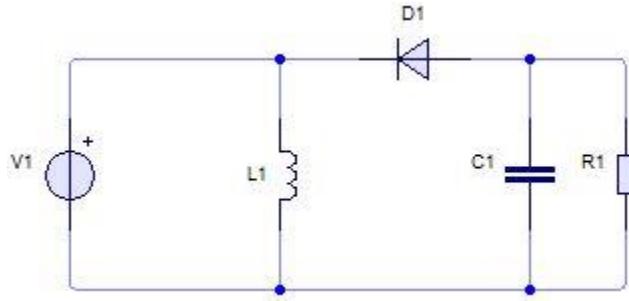
2.2.3 Buck-Boost Converter

Buck-boost converter adalah sebagai salah satu regulator mode pensaklaran yang menghasilkan tegangan keluaran yang lebih kecil atau lebih besar dari tegangan masukan. Berikut adalah gambar rangkaian *buck-boost converter* secara umum dapat dilihat pada Gambar 2.9:



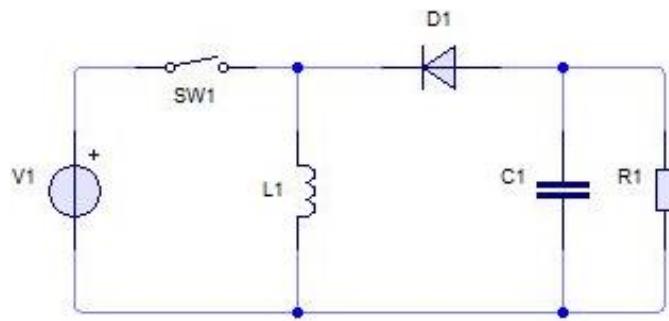
Gambar 2. 9 Rangkaian *Buck-Boost Converter* [7]

Prinsip kerja rangkaian ini dibagi menjadi 2 mode. Selama mode 1, transistor Q1 di-on-kan dan dioda mendapat bias mundur arus *input*, yang bertambah mengalir melalui induktor L dan transistor Q1. Selama mode 2, transistor Q1 di-off-kan. Dan arus mengalir melalui induktor L, diteruskan ke dioda dan ke beban. Energi yang tersimpan di dalam induktor L akan ditransfer ke beban. Dan arus induktor akan berkurang sampai transistor Q1 di-on-kan lagi pada siklus berikutnya [7].



Gambar 2. 10 Rangkaian *Buck-Boost* Saklar Tertutup [7].

Pada Gambar 2.10 menunjukkan rangkaian *buck-boost converter* dalam keadaan mosfet Q1 on. Hal ini menyebabkan dioda bekerja *reverse* sehingga arus akan mengalir ke induktor L. Dengan adanya arus yang mengalir ke induktor maka terjadi pengisian arus pada induktor sehingga arus induktor (IL) naik.

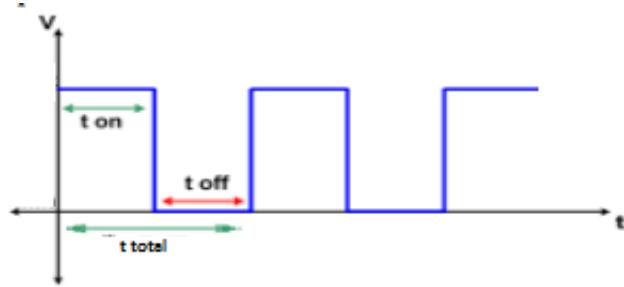


Gambar 2. 11 Rangkaian *Buck-Boost Converter* Saklar Terbuka

Pada Gambar 2.11 menunjukkan rangkaian *buck-boost converter* dalam keadaan mosfet Q1 off. Hal ini menyebabkan dioda bekerja *forward* sehingga arus mengalir L, C, D1 dan beban. Energi yang tersimpan di induktor mengalami *discharging*. *Buck-boost converter* menghasilkan tegangan keluaran yang terbalik tanpa memerlukan trafo dan menghasilkan tegangan keluaran yang lebih rendah maupun lebih tinggi dari tegangan masukan. Dengan catatan, bila *duty cycle* PWM sebagai penyulut *switch* lebih dari 50%, maka tegangan keluaran akan lebih tinggi dari tegangan masukan. Dan bila *duty cycle* PWM kurang dari 50%, maka tegangan keluaran akan lebih rendah dari tegangan masukan.

2.3 Pulse Width Modulation

Pengaturan lebar pulsa modulasi atau PWM merupakan salah satu teknik yang ampuh yang digunakan dalam sistem kendali (*control system*). Pengaturan lebar modulasi dipergunakan di berbagai bidang yang sangat luas, salah satu diantaranya adalah: *speed control* (kendali kecepatan), *power control* (kendali sistem tenaga), *measurement and communication* (pengukuran atau instrumentasi dan telekomunikasi) . Modulasi lebar pulsa (PWM) dicapai/diperoleh dengan bantuan sebuah gelombang kotak yang mana siklus kerja (*duty cycle*). PWM pada dasarnya adalah menyalakan (on) dan mematikan (off). Kuncinya adalah mengatur berapa lama waktu on dan off. [8]



Gambar 2. 12 Gelombang Pulse Width Modulation

Timer on (t_{on}) adalah waktu dimana tegangan keluaran berada pada posisi tinggi (baca: *high* atau 1) dan, timer off (t_{off}) adalah waktu dimana tegangan keluaran berada pada posisi rendah (baca: *low* atau 0). Anggap t_{total} adalah waktu satu siklus atau penjumlahan antara t_{on} dengan t_{off} , biasa dikenal dengan istilah “periode satu gelombang”.

$$t_{total} = t_{on} + t_{off} \dots \dots \dots \quad (2.1)$$

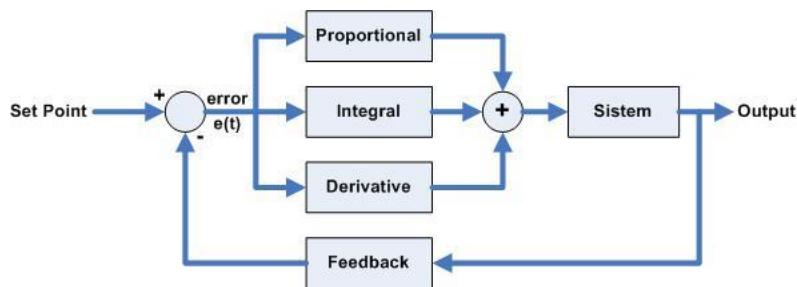
Tegangan keluaran dapat bervariasi dengan *duty-cycle* dan dapat dirumuskan sebagai berikut,

$$V_{out} = t_{total} \times V_{in} \dots \dots \dots \quad (2.2)$$

Dari rumus diatas dapat ditarik kesimpulan bahwa tegangan keluaran dapat diubah-ubah secara langsung dengan mengubah nilai t_{on} . Apabila t_{on} adalah 0, V_{out} juga akan 0. Apabila t_{on} adalah T_{total} maka V_{out} sama dengan V_{in} [8].

2.4 Kontrol PID (*Proportional–Integral–Derivative*)

Kontroler PID adalah hasil gabungan dari 3 bentuk kontroler yaitu kontroler *proporsional*, *integral* dan *derivative*. Biasanya kontrol ini digunakan di dunia industri [9]. Kontrol P (*Proportional*), I (*Integral*) dan D (*Derivative*) dapat digunakan bersamaan secara paralel atau bisa juga digunakan secara terpisah. Contohnya bisa kontrol P (*Proportional*) atau I (*Integral*) atau D (*Derivative*) saja. Bisa juga menggunakan kontrol P-I (*Proportional–Integral*) saja. Pada Gambar 2.13 merupakan blok diagram dari Kontrol PID.



Gambar 2. 13 Blok Diagram kontrol *Proportional Integral Derivative* (PID) [9]

Dari Gambar 2.13 bahwa *set point* yang dimaksud adalah nilai besaran yang diinginkan. *Feedback* adalah mekanisme pengaturan terhadap suatu sistem fisik yang dilakukan sedemikian rupa sehingga mekanisme yang berusaha menyesuaikan dengan nilai sistem yang dikendalikan. Sedangkan *error* adalah nilai kesalahan yang dihasilkan oleh suatu sistem

2.4.1. Kontrol P (*Proportional*)

Kontrol *Proportional* memiliki *output* sebanding dengan nilai konstanta Proportionalnya. Semakin besar nilai konstanta yang diberikan maka nilai *output* yang dikeluarkan akan besar juga. Dalam hal ini nilai kesalahan (selisih antara *setpoint* dan nilai aktualnya) dikali dengan konstanta proportionalnya. Persamaan kontrolnya dirumuskan sebagai berikut [9].

$$U_t = K_p e(t) \dots \dots \dots \quad (2.3)$$

Dengan U_t merupakan nilai keluaran terhadap waktu, K_p merupakan penguat (*gain*) dari *proportional* dan $e(t)$ merupakan nilai kesalahan yang terjadi [9].

2.4.2. Kontrol I (*Integral*)

Kontrol *Integral* berfungsi untuk mengurangi nilai *steady-state* eror menjadi nol. Kontrol *Integral* merupakan penjumlahan nilai kesalahan sebelumnya dengan nilai kesalahan yang terjadi saat ini dikali dengan Konstanta *Integral*. Persamaan 2.4 merupakan rumus dari kontrol I (*Integral*).

$$U_t = K_i \int_0^t e(t) dt \dots \quad (2.4)$$

Dengan U_t merupakan nilai keluaran terhadap waktu, K_i merupakan penguat (*gain*) dari integral dan $e(t)$ merupakan nilai kesalahan yang terjadi [9].

2.4.3. Kontrol D (*Derivative*)

Besar *output* dari Kontrol *Derivative* sama halnya dengan prinsip deferensial (turunan pada umumnya). Kontrol *Derivative* ini menggunakan perubahan kecepatan sinyal kesalahan sebagai parameter kontrol. Persamaan 2.5 merupakan rumus dari kontrol D (*Derivative*).

$$U_t = K_d \frac{de}{dt} et(dt) \dots \quad (2.5)$$

Dengan U_t merupakan nilai keluaran terhadap waktu, K_d merupakan penguat (*gain*) dari derivative dan $e(t)$ merupakan nilai kesalahan yang terjadi [9].

Sehingga dari ketiga konstanta dapat digabungkan sehingga dapat dirumuskan nilai keluaran dari PID dapat dilihat pada Persamaan 2.6.

$$U_t = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de}{dt} \dots \quad (2.6)$$

Dengan U_t adalah nilai keluaran kontrol, K adalah penguatan, $e(t)$ adalah sinyal kesalahan yang diperoleh dari selisih antara nilai sinyal *output* dan sinyal *setpoint* (sinyal yang diberikan). Parameter utama dari pengendali PID adalah penguatan *proportional* (K_p), penguat *integral* (K_i), dan penguatan *derivative* (K_d).

Dari persamaan diatas dapat disimpulkan bahwa nilai keluaran dari kontrol PID ini merupakan penjumlahan dari penguatan ketiga konstanta tersebut dengan dipengaruhi nilai kesalahan terhadap waktu tertentu.

BAB III

METODE PELAKSANAAN

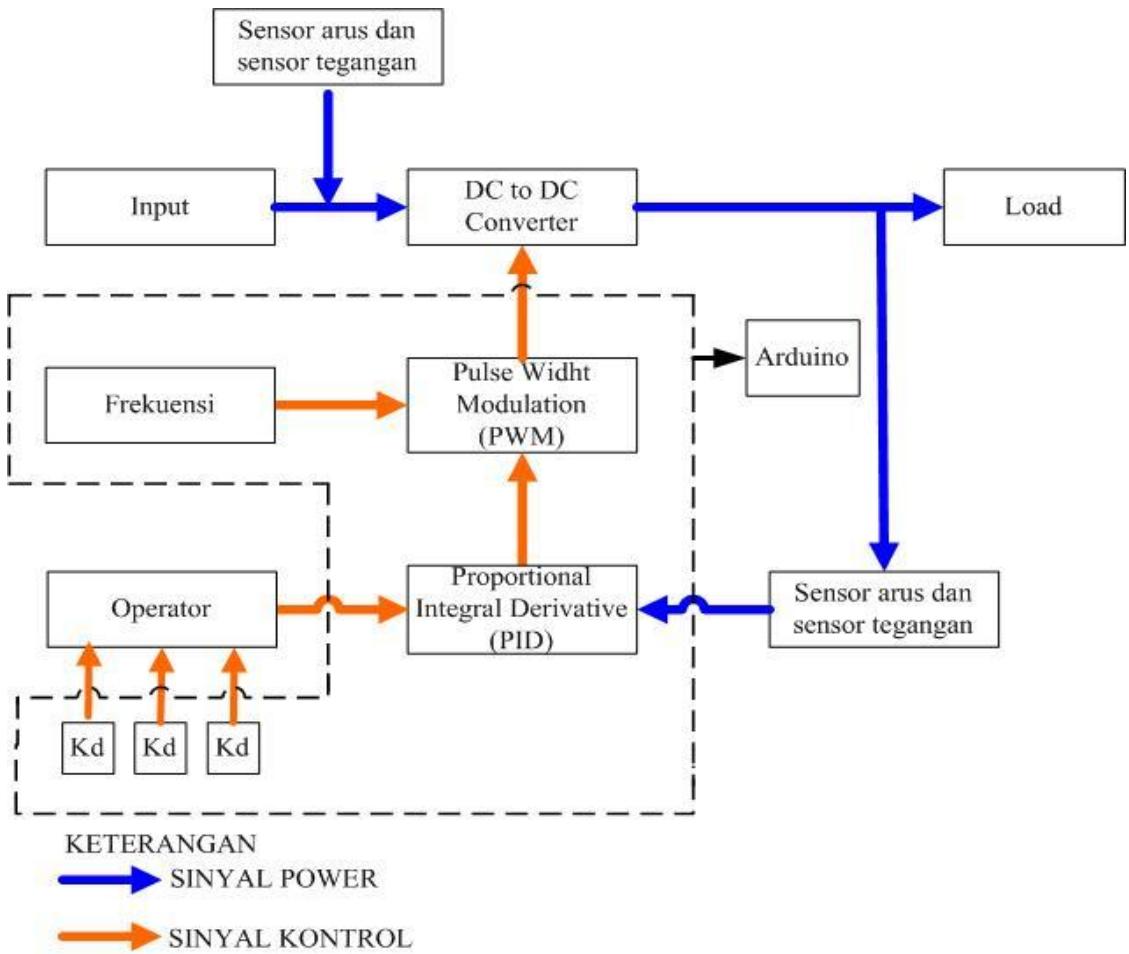
Pada Bab ini akan dibahas secara detail mengenai metode pelaksanaan pembuatan tugas akhir yang berjudul “*Programmable DC to DC Converter*”. Parameter-parameter yang penting dalam pembuatan alat *Programmable DC to DC Converter*” ini antara lain.

1. Tegangan *output* dapat naik atau turun dari tegangan *input* yang diberikan.
2. Tegangan *output* yang dihasilkan dapat berubah-ubah atau variable sesuai dengan keinginan operator.
3. Harus dapat menghasilkan tegangan *output* yang diinginkan secara otomatis.

Sehingga *design* dari peralatan yang dibuat terdiri dari.

1. *Buck-boost Converter* yang berfungsi untuk menaikkan atau menurunkan tegangan *output*.
2. Sensor tegangan digunakan untuk mengukur tegangan *output* apakah sesuai dengan tegangan yang diinginkan. Serta sensor arus untuk mengukur arus yang mengalir pada rangkaian.
3. Metode pengontrolan seperti P-I-D (*Proporsional Integral Derivative*) untuk menghasilkan tegangan *output* yang diinginkan.

Dari *design* diatas maka dapat digambarkan dalam bentuk blok diagram yang dapat dilihat pada Gambar 3.1



Gambar 3. 1 Blok diagram *Programmable DC to DC Converter*

Berdasarkan Gambar 3.1 *Input* yang digunakan untuk mensupply *DC to DC Converter* adalah *power supply* yang nilai tegangannya tetap yaitu 12 volt. Yang kemudian *converter* akan menurunkan atau menaikan tegangan *input* menjadi tegangan keluar sesuai dengan keinginan. Tegangan keluar yang keluar dari *converter* tersebut kemudian mengalir ke beban. Tegangan yang keluar dari *converter* akan menghasilkan eror-eror yang kemudian eror tersebut akan di kontrol oleh PID yang dikontrol oleh kontroler yaitu Arduino. Kemudian PID akan secara otomatis mengontrol PWM yang sesuai dengan tegangan yang diinginkan.

3.1 Komponen yang digunakan

Sesuai dengan parameter-parameter penting dan design dalam pembuatan alat *Programmable DC to DC Converter* sehingga adapun komponen yang digunakan diantaranya :

Tabel 3. 1 Komponen yang digunakan

No	Nama Komponen	Spesifikasi
1	<i>DC to DC Converter</i>	a. <i>Buck-Boost Converter</i> b. Tegangan keluar 0 sampai 28V c. Tegangan <i>input</i> 5V sampai 38V d. 400KHz <i>switching Frequency</i> e. <i>Current</i> 2A f. Daya Maksimal 12Watt
2	Sensor Arus INA219	a. <i>Power Supply</i> , 3V to 5V
3	Sensor Tegangan	a. <i>Max Voltage</i> 40V a. Tegangan <i>Input</i> 7V-12V b. <i>Input</i> dan <i>Ouput</i> (I/O) terdapat pada Pin 0 sampai Pin 13
4	Arduino UNO	c. I/O PWM Digital terdapat 6 Pin d. DC <i>Current</i> per I/O Pin 20mA e. DC <i>Current</i> for 3.3V Pin 50mA f. <i>Analog Input</i> Pin terdapat 6 Pin a. <i>Number of Character</i> 20 char x 4
5	<i>Liquid Crystal Display</i> (LCD)	<i>line</i> b. <i>Dimension</i> 98 x 60 x 13.6 mm c. <i>Driver</i> HD44780/sejenisnya
6	<i>Keypad</i> 4x4	a. Terdapat 4 kolom dan 4 baris b. Maksimum <i>rating</i> 24VDC, 30mA

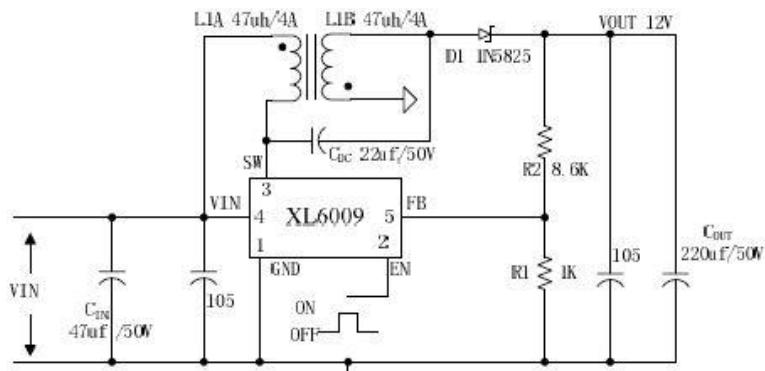
BAB IV

PEMBAHASAN

Pada bab ini akan dibahas mengenai proses pembuatan *Programmable DC to DC Converter* yang terdiri dari perancangan sistem kontrol, perancangan dan pembuatan alat dan program, serta pengujian sistem. Berikut adalah penjelasannya.

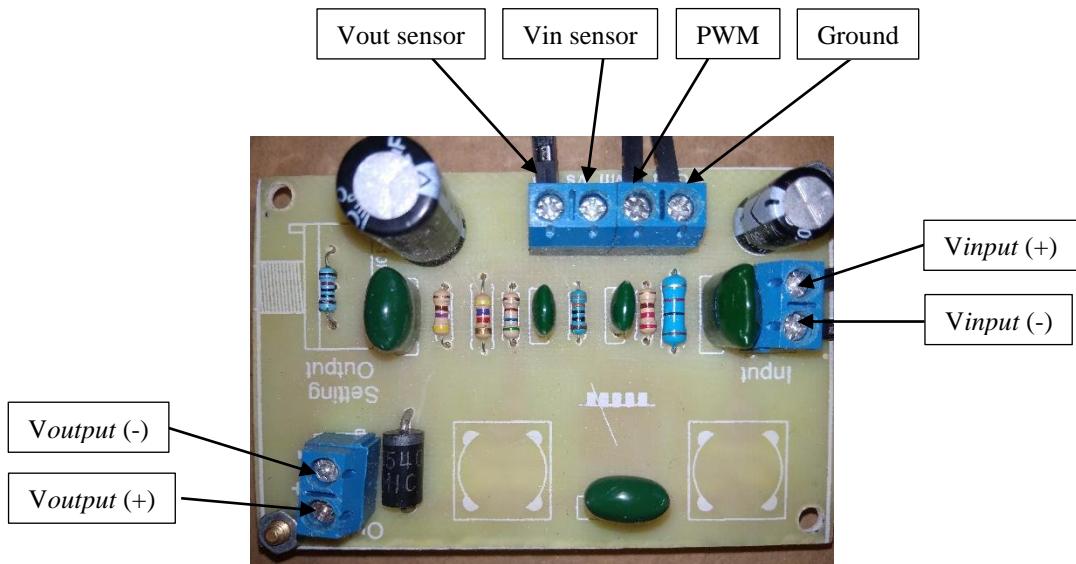
4.1 DC to DC Converter

Pada tugas akhir ini, *DC to DC Converter* yang digunakan adalah *DC to DC Converter* XLSEMI 6009. *Converter* ini tipe *Buck-Boost converter*. Pembuatan *DC to DC Converter* ini dibuat sama dengan modul XLSEMI 6009, dimulai dari komponen hingga rangkaian. Tujuan menduplikasi *Converter* ini agar mudah memasukan nilai PWM kedalam Pin 2 (Pin Enable) pada *converter*. Berikut adalah gambar rangkaian *schematic* dari *Buck-Boost Converter*. Dapat dilihat pada Gambar 4.1 sebagai berikut.



Gambar 4. 1 Rangkaian Schematic Buck-Boost Converter XLSEMI 6009

Berikut adalah *Hardware* dari rangkaian *Converter* XLSEMI 6009 dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Converter XLSEMI 6009

Pengujian pada *DC to DC Converter* ini dilakukan dengan menguji salah satu Pin pada *converter* yaitu Pin Enable. Pin Enable berfungsi untuk mematikan serta menghidupkan *converter*. Pengujian Pin enable ini menggunakan Arduino UNO. Pada pengujian ini dilakukan untuk mengetahui tegangan yang keluar apabila pada pin enable ini diberikan nilai PWM. Pin PWM yang digunakan yaitu Pin 6 dengan frekuensi 60 Hz. Berikut adalah *list* program dari pengujian.

```

int pwm=6
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  int data=analogWrite(pwm,0);
  delay(100);
}
  
```

Peletakan identitas PIN 6 PWM yang akan digunakan

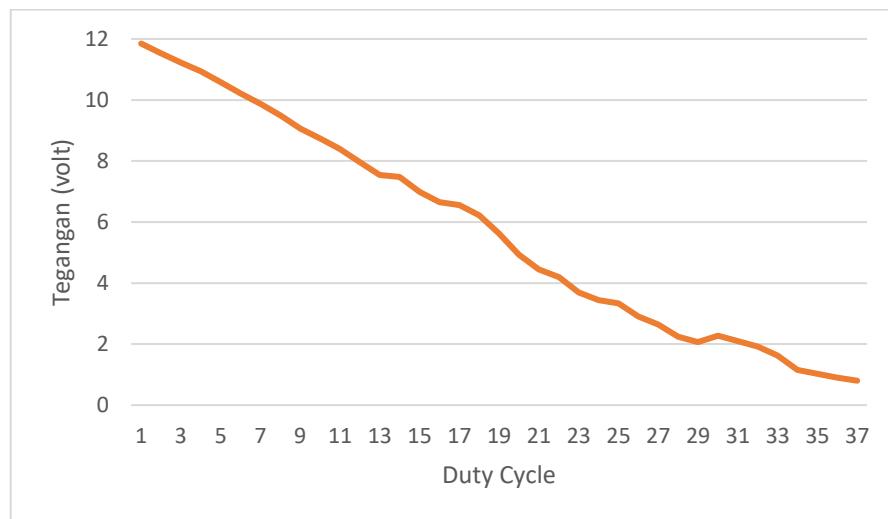
Setting baudrate yang akan ditampilkan pada serial monitor

Pembacaan PWM

Pengujian dilakukan dengan cara memasukan nilai PWM dari 1 sampai dengan 255, kemudian mengukur *Vout* pada *converter* yang diukur menggunakan

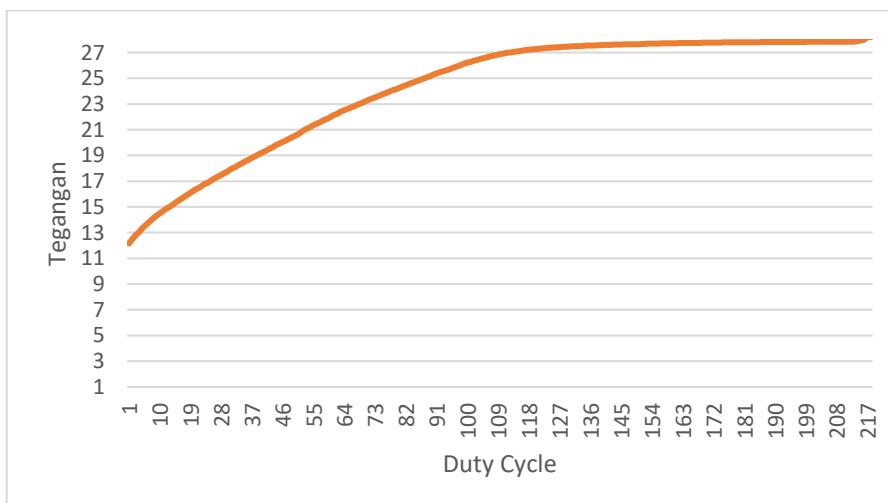
multitester. Dari pengujian yang dilakukan didapatkan hasil yang dapat dilihat pada tabel sebagai berikut.

Dari data yang diambil, pada Gambar 4.3 merupakan grafik yang menunjukan bahwa converter dapat menurunkan tegangan dari tegangan source yaitu 12 volt.



Gambar 4. 3 Grafik *Buck converter*

Dari Gambar 4.3 diatas maka dapat disimpulkan bahwa tegangan *set point* yaitu 12 volt dapat menurunkan tegangan hingga 0 volt tengan nilai pwm dari 0-37. Besar nilai *Duty cycle* untuk *Buck converter* adalah 14,5 %.



Gambar 4. 4 Grafik *Boost Converter*

Dari Gambar 4.4 diatas dapat dianalisa bahwa mulai dari nilai pwm 38 sampai 255 menunjukan bahwa converter ini dapat menaikan tegangan sampai 28 v dengan duty cycle sebesar 85.5 %.

Dari pengujian keseluruhan yang dilakukan didapatkan data berupa grafik yang dapat dilihat pada Gambar 4.5 sebagai berikut

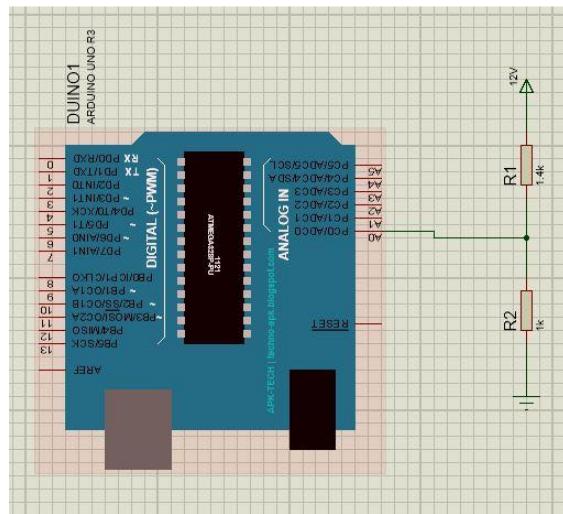


Gambar 4. 5 Grafik Pengujian Pin Enable.

Dari Gambar 4.5 diatas dapat dianalisa bahwa, dengan memberikan nilai PWM dari 1 sampai dengan 255 menghasilkan nilai yang relatif stabil. Secara keseluruhan semakin besar nilai PWM yang diberikan maka tegangan *output* yang dihasilkan juga akan semakin meningkat secara signifikan. Dan nilai *ripple* yang dihasilkan juga kecil. Hal ini dikarenakan frekuensi pada Pin 6 pada Arduino UNO sesuai.

4.2 Sensor Tegangan

Pada tugas akhir ini sensor yang digunakan adalah pembagi tegangan. Rangkaian pembagi tegangan berfungsi untuk mendeteksi tegangan yang keluar dari *DC to DC Converter*.



Gambar 4. 6 Rancangan Sensor Tegangan

Pada Gambar 4.6 merupakan rangkaian pembagi tegangan yang nantinya akan mendeteksi tegangan yang keluar dari *converter*. Tegangan keluaran dari rangkaian pembagi tegangan tersebut akan masuk ke Pin *analog* pada arduino (A0). Perhitungan rangkaian pembagi tegangan dapat dilihat pada persamaan berikut.

$$V_{out} = \frac{R_2}{R_1+R_2} X V_{in} \dots \quad (4.1)$$

Untuk menghitung nilai resistansi (R_1) dapat dilihat pada persamaan berikut.

$$5V = \frac{1K}{R_1+1K} X 12V \dots \dots \dots \quad (4.3)$$

$$R1 + 1K = \frac{12K}{5V} \dots \dots \dots \quad (4.4)$$

$$R1 + 1K = 2400 \dots \dots \dots \quad (4.5)$$

$$R1 = 2400 - 1000 \dots \quad (4.6)$$

$$B_1 \equiv 1400 \quad (4.7)$$

Jadi, nilai resistor satu (R_1) dan resistor dua (R_2) untuk sensor tegangan adalah 1,4K ohm dan 1K ohm.

Untuk pengujian sensor tegangan ini dilakukan dengan cara membandingkan tegangan yang keluar dari sensor dengan tegangan *output* yang terukur menggunakan multimeter dengan tegangan *source converter* 12 Volt. Berikut adalah tabel pengujian dari sensor tegangan yang dapat dilihat pada Tabel 4.1

Tabel 4. 1 Pengujian Sensor Tegangan

Set Point	Pengukuran		
	Sensor tegangan	Multimeter	Eror
1 Volt	1,09 Volt	1,05 Volt	3,81%
2 Volt	2,11 Volt	2,08 Volt	1,44%
3 Volt	3,15 Volt	3,07 Volt	2,61%
4 Volt	4,14 Volt	4,17 Volt	0,72%
5 Volt	4,96 Volt	5,01 Volt	1,00%
6 Volt	5,89 Volt	6,03 Volt	2,32%
7 Volt	7,16 Volt	7,13 Volt	0,42%
8 Volt	8,13 Volt	8,08 Volt	0,62%
9 Volt	8,95 Volt	9,02 Volt	0,78%
10 Volt	10,11 Volt	10,16 Volt	0,49%
11 Volt	11,2 Volt	11,11 Volt	0,81%
12 Volt	12,09 Volt	12,03 Volt	0,50%
13 Volt	12,93 Volt	13,01 Volt	0,61%
14 Volt	14,06 Volt	14,1 Volt	0,28%
15 Volt	15,2 Volt	15,03 Volt	1,13%
16 Volt	15,96 Volt	16,1 Volt	0,87%
17 Volt	17,21 Volt	17,06 Volt	0,88%
18 Volt	18,16 Volt	18,08 Volt	0,44%
19 Volt	19,23 Volt	19,2 Volt	0,16%
20 Volt	19,88 Volt	20,06 Volt	0,90%
21 Volt	21,23 Volt	21,03 Volt	0,95%
22 Volt	22,06 Volt	22,17 Volt	0,50%
23 Volt	23,02 Volt	23,08 Volt	0,26%
24 Volt	24,17 Volt	24,17 Volt	0,00%
25 Volt	24,95 Volt	25,07 Volt	0,48%
26 Volt	26,13 Volt	26,09 Volt	0,15%
27 Volt	27,18 Volt	27,05 Volt	0,48%
28 Volt	28,04 Volt	28,12 Volt	0,28%

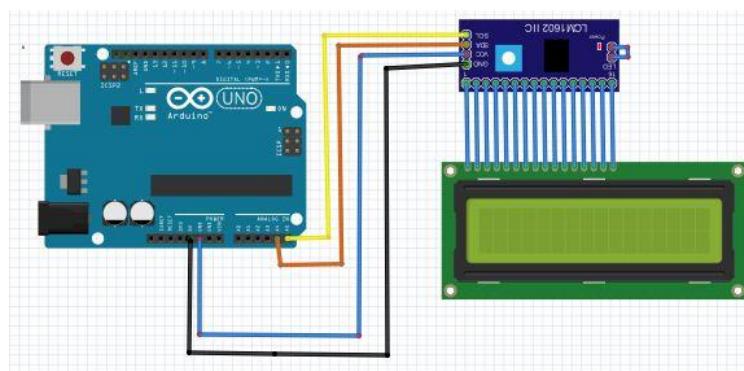
Untuk menghitung nilai eror yang dihasilkan suatu instrumentasi dapat dirumuskan pada persamaan berikut.

$$\text{Error} = \left| \frac{\text{Sensor - Multimeter}}{\text{Sensor}} \times 100 \right| \dots \quad (4.8)$$

Dari data table diatas dapat disimpulkan bahwa keluaran dari sensor tegangan tidak linierisasi dikarenakan pengaruh pada frekuensi yang disediakan oleh arduino sebesar 60Hz sedangkan *switching frequency* pada converter XLSEMI 6009 sebesar 400KHz.

4.3 *Liquid Crystal Display (LCD)*

LCD yang digunakan adalah LCD 20 X 4 . LCD adalah sebuah *display* untuk menampilkan data. Untuk serial komunikasi sebagai komponen tambahan untuk menampilkan data ini menggunakan I2C (*Inter Integrated Circuit*) agar menghemat Pin yang akan digunakan. Karena apabila tidak menggunakan I2C maka Pin yang digunakan akan banyak. I2C (*Inter Integrated Circuit*) adalah komunikasi serial dua arah yang menggunakan dua saluran yang digunakan untuk mengirim dan menerima data ke arduino. Pada I2C ada 4 Pin yaitu Pin Vcc, Pin Ground, Pin SDA dan Pin SCL. Pin SDA berfungsi untuk menghantarkan sinyal *clock* sedangkan Pin SCL berfungsi untuk mentransaksikan data. Pemasangan wirring kabel ke arduino Pin Vcc akan dihubungkan ke 5 Volt arduino, Pin Ground dihubungkan pada Ground, Pin SDA dihubungkan ke Pin A4 arduino, dan Pin SCL dihubungkan ke A5. LCD ini nantinya akan menampilkan setting PID, tegangan yang diinginkan, serta tegangan yang keluar. Untuk lebih jelas rangkaian LCD dilihat pada Gambar 4.7 sebagai berikut.



Gambar 4. 7 Rangkaian *Liquid Crystal Display*(LCD)

Pengujian pada rangkaian LCD ini menggunakan program arduino dengan *listing* program sebagai berikut.

```
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
#define LCD_COL 20
#define LCD_ROW 4

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
boolean backlight = BACKLIGHT_ON;

void setup()
{
    lcd.begin(LCD_COL,LCD_ROW);
}
void loop()
{
    lcd.setCursor(3,0);
    lcd.print("DC-DC CONVERTER");
    delay(1000);
}
```

Library untuk *Liquid Crystal Display* (LCD)

Inisialisasi LCD

Untuk menampilkan karakter “DC-DC CONVERTER” pada kolom ke-3 dan baris ke 0

Dari program pengujian pada *Liquid Cristal Display* tampilan dapat dilihat pada Gambar 4.8.



Gambar 4. 8 Tampilan Pengujian *Liquid Crystal Display* (LCD)

Terdapat beberapa menu yang disediakan yang berfungsi untuk mempermudah operator/user dalam memilih menu yang diinginkan. Antara lain Menu *Keypad*. Didalam menu *Keypad* ini terdapat 2 menu lagi didalamnya antara lain menu *default* dan menu *set PID*. Menu-menu tersebut memiliki fungsi masing-masing. Berikut adalah *list* program keseluruhan menu

```

#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
#define LCD_COL 20
#define LCD_ROW 4
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
boolean backlight = BACKLIGHT_ON;

String Menu_Mode[2]=
{
    "1. default",
    "2. set PID"
};
String Menu_acces[2]=
{
    "1. Keypad",
};
String Menu_Metode[2]=
{
    "1. parallax",
    "2. Matlab"
};

void setup()
{
lcd.begin(LCD_COL,LCD_ROW);
lcd.setBacklight(backlight);
}

void loop()
{
while(langkah==0)
{
    lcd.print(" Choose acces ");
    lcd.setCursor(0,1);
    lcd.print(" ===== ");
    lcd.setCursor(0,2);
    lcd.print(Menu_acces[up]);
}
while(langkah==999)
{
    lcd.setCursor(0,0);
    lcd.print(" Choose Metode");
    lcd.setCursor(0,1);
    lcd.print(" ===== ");
    lcd.setCursor(0,2);
    lcd.print(Menu_Metode[up1]);
}

```

Library untuk Liquid Crystal Display (LCD)

Tampilan menu pilihan pada LCD

Inisialisasi LCD

Tampilan utama pada LCD

Tampilan metode

```

while(program==1)
{
lcd.setCursor(14,0);
lcd.print(teg);
lcd.setCursor(0,0);
lcd.print("Set Parameter");
lcd.setCursor(0,1);
lcd.print(" ======");
lcd.setCursor(0,2);
lcd.print("Kp= ");
lcd.setCursor(0,3);
lcd.print("Ki= ");
lcd.setCursor(10,2);
lcd.print("Kd= ");
lcd.setCursor(10,3);
lcd.print("Sp=");
}

```

Tampilan untuk mengatur atau memasukan nilai Konstanta PID dan Set point

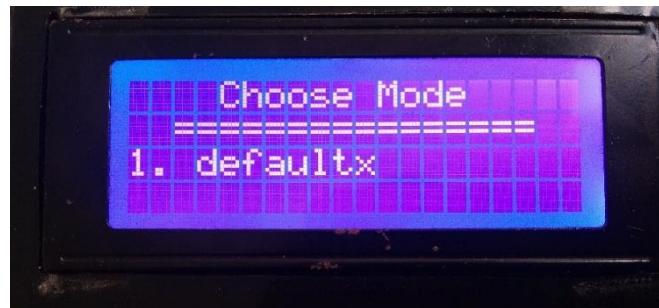
Dari program yang telah dibuat maka tampilan *Hardware* dari *Liquid Crystal Display* (LCD) dapat dilihat pada Gambar 4.9. Pada Gambar 4.9 merupakan tampilan awal jika tombol On ditekan.



Gambar 4. 9 *Hardware Liquid Crystal Display* (LCD)

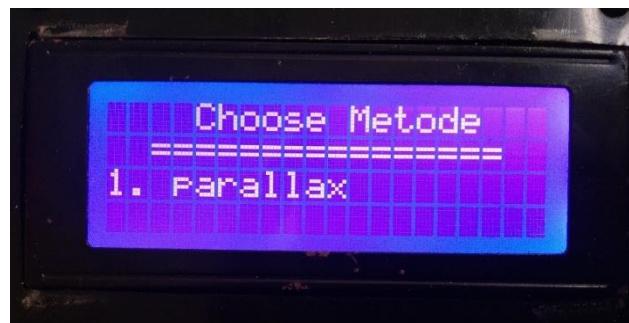
Selain itu pada Gambar 4.9 juga terdapat menu lain yang dapat dipilih antara lain.

- Menu *default*. Menu ini merupakan menu yang telah menyediakan nilai Kp, Ki dan Kd dengan respon yang bagus. Dapat dilihat pada Gambar 4.10



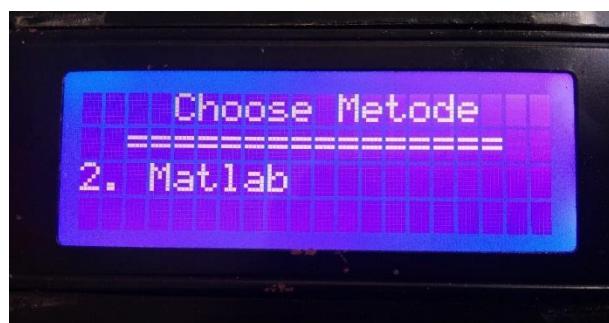
Gambar 4. 10 Tampilan dengan menu *default*

- Menu *parallax*. Menu ini merupakan menu yang menampilkan metode untuk melihat grafik yang respon hingga mencapai *set point* yang diinginkan. Dengan menggunakan serial komunikasi *Parallax*. Tampilannya dapat dilihat pada Gambar 4.11



Gambar 4. 11 Tampilan Menu *Parallax*

- Menu *Matlab*. Menu ini merupakan menu yang menampilkan metode untuk melihat grafik yang respon hingga mencapai *set point* yang diinginkan. Dengan menggunakan serial komunikasi *Matlab*. Tampilannya dapat dilihat pada Gambar 4.12



Gambar 4. 12 Tampilan Menu *Matlab*

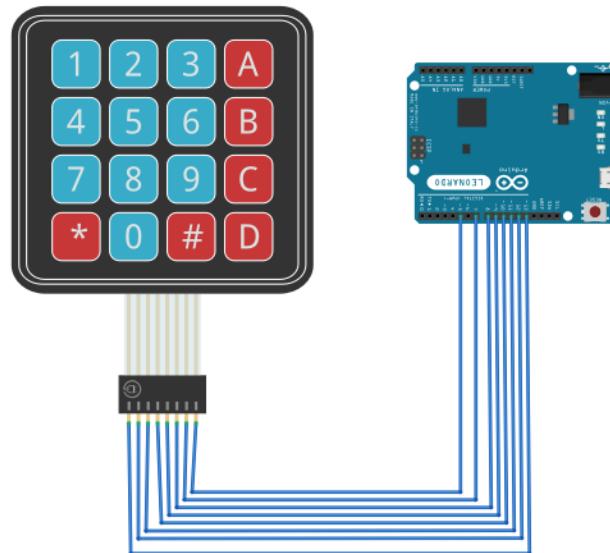
- Menu *Set Parameter*. Menu ini berfungsi untuk menentukan nilai Kp, Ki dan Kd yang diinginkan. Tampilan dapat dilihat pada Gambar 4.13



Gambar 4. 13 Tampilan Menu *Set Parameter*

4.4 Keypad

Keypad berfungsi untuk memasukan data besarnya nilai Konstanta Proportional, Kinstansa Integral, Konstantsa Derivative serta nilai tegangan yang diinginkan. *Keypad* memiliki 8 Port *output* yang nantinya akan dimasukan ke PIN arduino. Ada 8 PIN pada *Keypad* adalah R1, R2, R3 dan R4 untuk baris dan C1, C2, C3 dan C4 untuk kolom. *Keypad* dirancang untuk memasukan nilai tegangan yang diinginkan serta Konstanta dari PID yang diinput secara manual. Untuk lebih jelas dapat dilihat pada Gambar 4.14.



Gambar 4. 14 Perancangan dari *Keypad*

Pada *Keypad* ada beberapa menu yang dibuat. Dapat dilihat pada Gambar 4.13.



Gambar 4. 15 Menu-menu yang disediakan

- Ket :
1. Pembagi bilangan 10
 2. Pembagi bilangan 100
 3. Pembagi bilangan 1000
 4. Enter

Pengujian pada rangkaian *Keypad* ini menggunakan program arduino dengan *listing* program sebagai berikut.

```
#include <Keypad.h>
#define BINTANG 42
#define PAGAR 35
#define NOL 48
#define A 65
#define B 66
#define C 67
#define D 68
#define ROWS 4 //four rows
#define COLS 4
char keys[ROWS][COLS] =
{
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
}
```

Inisialisasi Library
Keypad

```

{'*','0','#','D'}
};

byte rowPins[ROWS] = {13,12,11,10}; //connect to the row pinouts of the Keypad
byte colPins[COLS] = {9,8,7,5};
Keypad customKeypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setkp()
{
    customKey = customKeypad.getKey();
    if(customKey >= '0' && customKey <= '9')
    {
        kp = kp * 10 + (customKey-'0');
    }
    if(customKey == 'A')
    {
        kp = kp/10;
        nilaikp = kp;
    }
    if(customKey == 'B')
    {
        kp = kp/100;
        nilaikp = kp;
    }
    if(customKey == 'C')
    {
        kp = kp/1000;
        nilaikd = kd;
    }
    if(customKey == '*')
    {
        nilaikp = kp;
        return;
    }
    setkp();
}
void setki()
{
    customKey = customKeypad.getKey();
    if(customKey >= '0' && customKey <= '9')
    {
        ki = ki * 10 + (customKey-'0');
    }
    if(customKey == 'A')
    {
        ki = ki/10;
        nilaiki = ki;
    }
    if(customKey == 'B')
    {
        ki = ki/100;
        nilaiki = ki;
    }
    if(customKey == 'C')
    {
        ki = ki/1000;
    }
}

```

Program fungsi untuk memasukan nilai konstanta Kp

Program fungsi untuk memasukan nilai konstanta Ki

```

        nilaiki = ki;
    }
    if(customKey == '*')
    {
        nilaiki = ki;
        return;
    }
    setki();
}
void setkd()
{
    customKey = customKeypad.getKey();
    if((customKey >= '0' && customKey <= '9'))
    {
        kd = kd * 10 + (customKey-'0');
    }
    if(customKey == 'A')
    {
        kd = kd/10;
        nilaikd = kd;
    }
    if(customKey == 'B')
    {
        kd = kd/100;
        nilaikd = kd;
    }
    if(customKey == 'C')
    {
        kd = kd/1000;
        nilaikd = kd;
    }
    if(customKey == '*')
    {
        nilaikd = kd;
        return;
    }
    setkd();
}
void setsp()
{
    customKey = customKeypad.getKey();
    if(customKey >= '0' && customKey <= '9')
    {
        sp = sp * 10 + (customKey - '0');
        lcd.setCursor(0,1);
    }
    if(customKey == '*')
    {
        keydd3 = sp;
        lcd.clear();
        return;
    }
    setsp();
}

```

Program fungsi untuk memasukan nilai konstanta Kd

Program fungsi untuk memasukan nilai Set point

```

void setsp1()
{
    customKey = customKeypad.getKey();
    if(customKey >= '0' && customKey <= '9')
    {
        sp = sp * 10 + (customKey - '0');
        lcd.setCursor(0,1);
    }
    if(customKey == '*'){
        keydd3 = sp;
        lcd.clear();
        return;
    }
    setsp1();
}

```

Program fungsi untuk memasukan nilai *Set point*

Program fungsi untuk memasukan nilai *Set point*

4.5 Pembuatan Box

Pembuatan *Box* ini dibuat untuk memudahkan dalam pembawaan dan tempat untuk meletakan komponen-komponen. Pembuatan *Box* ini dimulai dari pembuatan *box* yang berbentuk seperti balok yang didalamnya terdapat peletakan untuk komponen. Pemilihan bahan untuk *box* tersebut adalah *acrilic* yang ketebalannya 2mm.

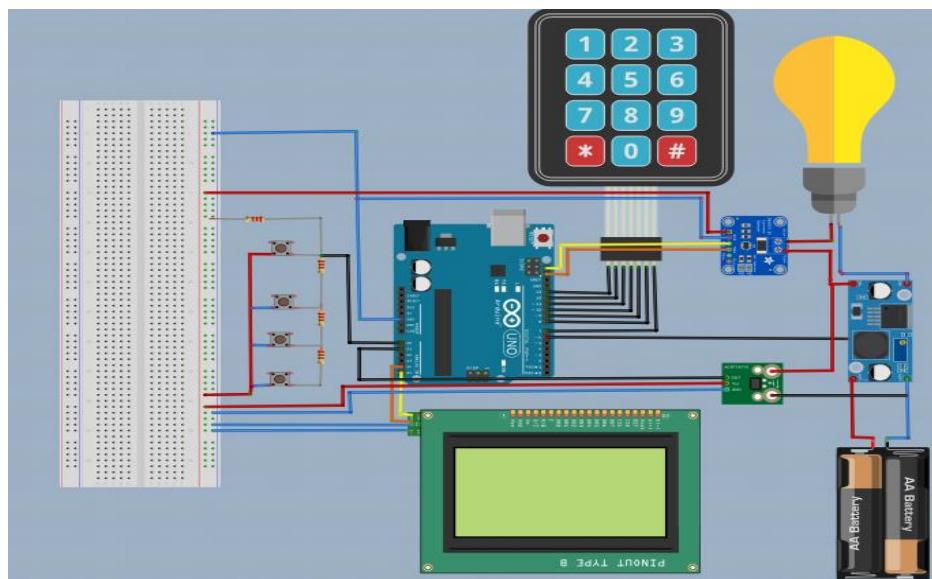


Gambar 4. 16 Hasil dan Desain *Box*

4.6 Rangkaian Kontrol

Rangkaian kontrol dirancang dalam sebuah *box* dengan bahan yang digunakan adalah *acrilic* dengan menggunakan *base* sebagai alas pada *box* tersebut. Didalam rangkaian tersebut terdapat Arduino Uno yang berfungsi untuk mengolah data yang dikirimkan oleh sensor tegangan dan sensor arus serta untuk mengontrol

Pulse Width Modulation dan *Proportional Integral Derivative*, Sensor tegangan dan sensor arus, *DC to Dc Converter* sebagai penurun dan penaik tegangan, LCD, *push button* dan *Keypad*. Gambar 4.13 merupakan rangkaian kontrol dari *Programmable DC to Dc Converter*.



Gambar 4. 17 Rangkaian Kontrol *Programmable DC to DC Converter*

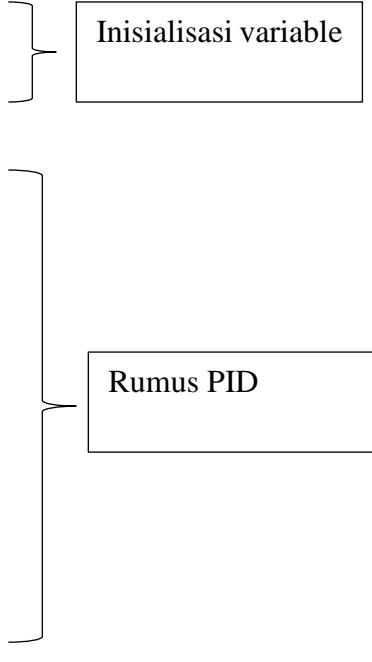
Dari rangkaian diatas dapat dijelaskan bahwa untuk mengontrol sistem menggunakan Arduino Uno. *Push button 1* (Pb1), *push button 2*, *push button 3*, *push button 4* ini akan dipasang pada Pin A0. Keempat *push button* ini mempunyai fungsi masing-masing. Antara lain Pb1 berfungsi untuk mengembalikan sistem ke menu awal. Pb2 berfungsi sebagai enter. Pb3 sebagai *step up* dan Pb4 sebagai *step down*. Ketika *set point* diberikan Arduino akan mengolah data yang diberikan dan kemudian akan secara otomatis mencari nilai PWM. PWM ini dimasukan pada Pin 6. Sehingga tegangan keluaran akan secara otomatis mengeluarkan tegangan yang sesuai dengan yang diinginkan atau *set point* yang telah diberikan.

4.7 Kontrol *Proportional* (P)

Kontrol *Proportional* ini lebih dikenal dengan sebutan *gain* (penguat). Perubahan nilai Konstanta *Proportional* (K_p) akan mempengaruhi bentuk

gelombang yaitu tinggi rendahnya *overshoot* yang dihasilkan oleh suatu sistem. Kontrol *Proportional* ini bertanggung jawab atas nilai kesalahan saat ini.

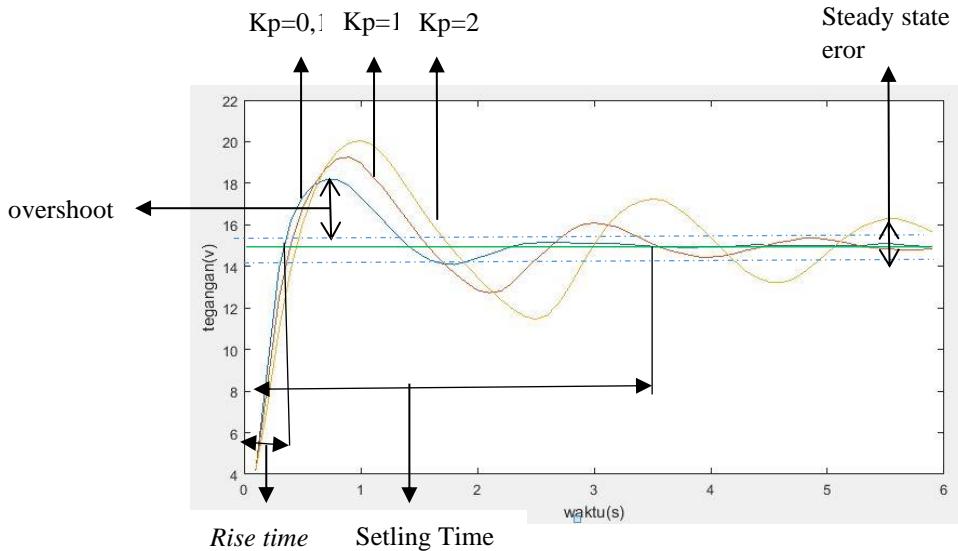
```
float kp,ki,kd,pid,  
float integral,proporsional,derivative;  
float error,kesalahan;  
  
void loop()  
{  
float hitungPID(float tegangan,float teg)  
{  
kesalahan=tegangan-teg;  
proporsional=kesalahan*kp;  
integral=integral+(error*ki);  
derivatif=(error-kesalahan)*kd;  
integral+=kesalahan;  
error=kesalahan;  
pid=(proporsional+integral+derivatif)/0.3;  
return (pid);  
}  
}
```



Dari program diatas dapat diuji untuk melihat grafik respon. Penentuan Nilai Kp menggunakan Metode *Trial and Error* dapat dilihat pada Tabel 4.2

Tabel 4. 2 Percobaan Metode *Trial and Error* Konstanta *Prortionial* (Kp)

Percobaan ke-	Nilai Konstanta Proportional(Kp)
1	0,1
2	1
3	2



Gambar 4. 18 Grafik Perbandingan nilai K_p

Dari Gambar 4.18 diatas dapat disimpulkan bahwa dilihat dari nilai *overshoot* (nilai yang melebih nilai *set point*) dari masing masing nilai K_p berbeda-beda. Untuk nilai $K_p=0,1$, presentase overshoot yaitu sebesar 20%. Untuk nilai $K_p=1$, presentase overshoot yaitu sebesar 26,6%. Dan Untuk nilai $K_p=2$, presentase overshoot yaitu sebesar 33,3%. Selain nilai overshoot, nilai *rise time* (Waktu yang dibutuhkan untuk mencapai nilai set point atau yang mendekati set point 10%) pada $K_p=0,1$ sebesar 0.2 s. Untuk nilai $K_p=1$ sebesar 0.3 s. Dan untuk nilai $K_p=2$ sebesar 0.4s. Setling time (waktu yang dibutuhkan untuk mencapai nilai stabil), jadi besar setling time $K_p=0,1$ sebesar 2s. Untuk $K_p=1$ sebesar 4s, sedangkan dengan nilai $K_p=2$ sebesar 6 s lebih. Serta nilai eror (nilai kesalahan yang terjadi) dapat dilihat dari grafik bahwa nilai $K_p=0,1$ relatif lebih kecil daripada nilai K_p yang lain. Dan nilai $K_p = 0,1$ juga sudah sangat stabil respon yang dihasilkan. *Set Point* yang diberikan sebesar 15 volt.

4.8 Kontrol Proporsional Integral (P-I)

Kontrol *proporsional integral* merupakan perubahan dari keluaran kontrol integral, berubah dengan fungsi waktu yang sebanding dengan sinyal kesalahan [10]. Kontrol *proporsional Integral* (U_t) merupakan gabungan antara kontrol *Proporsional* dan kontrol *Integral*, dimana nilai kesalahan (e_t) dikali dengan konstanta *proporsional* (K_p) ditambah dengan penjumlahan nilai eror sebelumnya

dan kesalahan saat ini dikali dengan konstanta *integral* (K_i). Berikut adalah *list Program* dari Pengontrolan *Proportional Integral*.

```

float kp,ki,kd,pid,
float integral,proporsional,derivative;
float error,kesalahan;

void loop()
{
float hitungPID(float tegangan,float teg)
{
    kesalahan=tegangan-teg;
    proporsional=kesalahan*kp;
    integral=integral+(error*ki);
    derivatif=(error-kesalahan)*kd;
    integral+=kesalahan;
    error=kesalahan;
    pid=(proporsional+integral+derivatif)/0.3;
    return (pid);
}
}

```

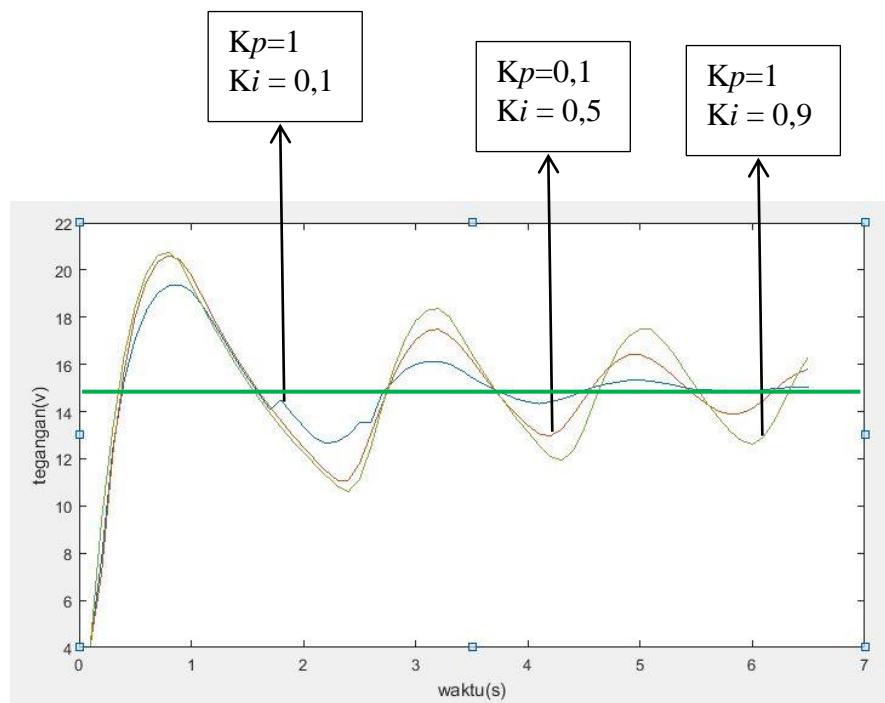
Inisialisasi variable

Rumus PID

Dari program diatas dapat diuji untuk melihat grafik respon yang dihasilkan. Penentuan Nilai K_p dan K_i menggunakan Metode *Trial and Error* dapat dilihat pada Tabel 4.3

Tabel 4. 3 Percobaan Metode *Trial and Error* Konstanta *Prortionial* (K_p) dan Konstanta Integral (K_i)

Percobaan ke-	Nilai Konstanta Proportional(K_p)	Nilai Konstanta Integral (K_i)
1	1	0,1
2	1	0,5
3	1	0,9



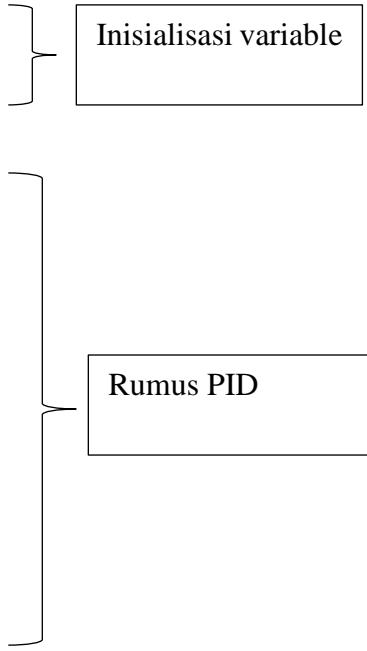
Gambar 4. 19 Grafik Perbandingan nilai K_p dan K_i

Dari Gambar 4.19 diatas dapat disimpulkan bahwa dilihat dari nilai *overshoot* (nilai yang melebih nilai *set point*) dari masing masing nilai K_p dan K_i berbeda-beda. Untuk nilai $K_p=1$ dan $K_i=0,1$ presentase overshoot yaitu sebesar 29,3%. Untuk nilai $K_p=0,1$ dan $K_i=0,5$, presentase overshoot yaitu sebesar 36,6%. Dan Untuk nilai $K_p=1$ dan $K_i=0,9$, presentase overshoot yaitu sebesar 37,3%. Selain nilai overshoot, nilai *rise time* (Waktu yang dibutuhkan untuk mencapai nilai set point atau yang mendekati set point 10%) pada $K_p=1$ dan $K_i=0,1$ sebesar 0,3 s. Untuk nilai $K_p=0,1$ dan $K_i=0,5$ sebesar 0.3 s. Dan untuk nilai $K_p=1$ dan $K_i=0,9$ sebesar 0,3s. Setling time (waktu yang dibutuhkan untuk mencapai nilai stabil), jadi besar setling time $K_p=1$ dan $K_i=0,1$ sebesar 5s. Untuk $K_p=0,1$ dan $K_i=0,5$ sebesar 6s lebih, sedangkan dengan nilai $K_p=1$ dan $K_i=0,9$ sebesar 6 s lebih. Serta nilai eror (nilai kesalahan yang terjadi) dapat dilihat dari grafik bahwa nilai $K_p=1$ dan $K_i=0,1$ relatif lebih kecil daripada nilai K_p dan K_i yang lainnya. Dan nilai $K_p = 1$ dan $K_i=0,1$ juga sudah sangat stabil respon yang dihasilkan. *Set Point* yang diberikan sebesar 15 volt.

4.9 Kontrol Proportional Integral Derivative

Kontrol *Proportional Integral Derivative* adalah gabungan dari Kontrol *Proportional*, Kontrol *Integral*, dan Kontrol *Derivative*. Pada tugas akhir ini PID difungsikan untuk mengurangi nilai eror yang terjadi. Metode yang digunakan pada tugas akhir ini adalah Metode *Trial and Error*. Metode *Trial and Error* adalah metode yang dilakukan dengan cara sistem coba-coba. Yang artinya nilai K_p , nilai K_i dan nilai K_d dimasukan pada sistem secara coba-coba. Hal ini dilakukan karena belum adanya teori yang dapat menentukan nilai K_p , nilai K_i dan nilai K_d . Pengujian yang dilakukan dengan *set point* 15 volt. Berikut adalah *list* program dari kontrol *Proportional Integral Derivative*.

```
float kp,ki,kd,pid,  
float integral,proporsional,derivative;  
float error,kesalahan;  
  
void loop()  
{  
float hitungPID(float tegangan,float teg)  
{  
    kesalahan=tegangan-teg;  
    proporsional=kesalahan*kp;  
    integral=integral+(error*ki);  
    derivatif=(error-kesalahan)*kd;  
    integral+=kesalahan;  
    error=kesalahan;  
    pid=(proporsional+integral+derivatif)/0.3;  
    return (pid);  
}  
}
```



Inisialisasi variable

Rumus PID

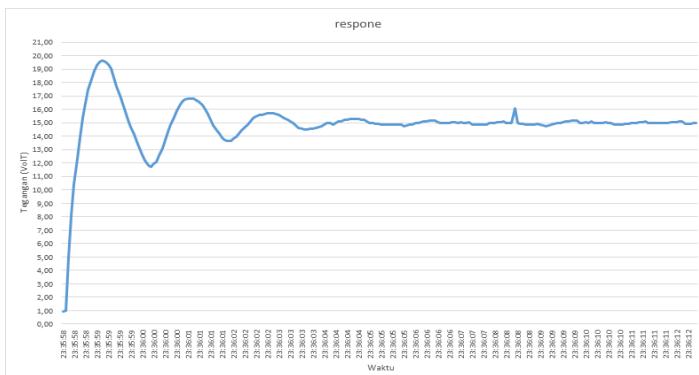
Berikut adalah hasil pengujian pengaruh *Proportional Integral Derivative* (PID) terhadap sistem.

Tabel 4. 4 Nilai K_p dan K_i tetap, sedangkan nilai K_d berubah-ubah

No	K_p	K_i	K_d
1	1	0,1	0,1
2	1	0,1	0,2
3	1	0,1	0,3
4	1	0,1	0,4
5	1	0,1	0,5

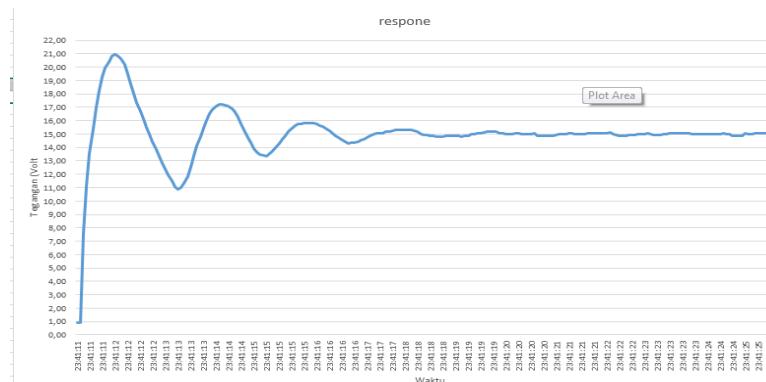
Dari tabel diatas untuk melihat grafik respon dapat dilihat pada Gambar 4.10 sampai dengan Gambar 4.19

1. $K_p=1$; $K_i=0,1$; $K_d=0,1$



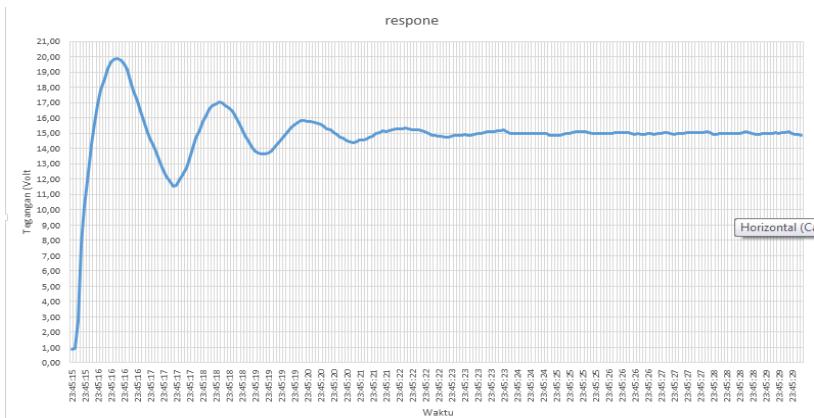
Gambar 4. 20 Grafik dengan nilai $K_p=1$; $K_i=0,1$; $K_d=0,1$

2. $K_p=1$; $K_i=0,1$; $K_d=0,3$



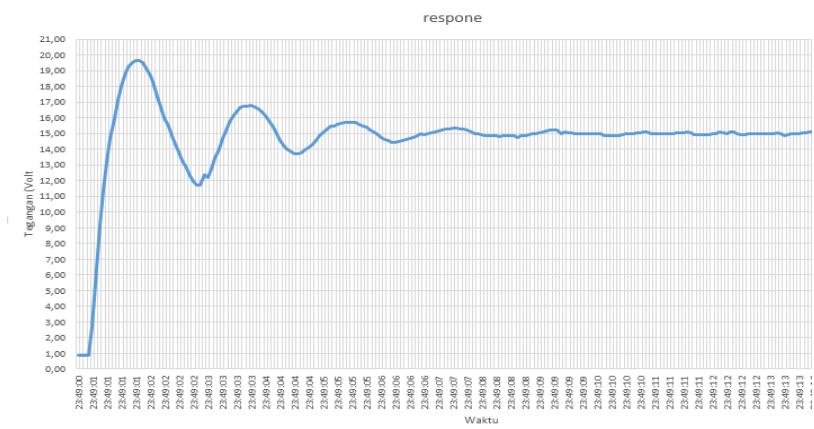
Gambar 4. 21 Grafik dengan nilai $K_p=1$; $K_i=0,1$; $K_d=0,3$

3. $K_p=1$; $K_i=0,1$; $K_d=0,5$



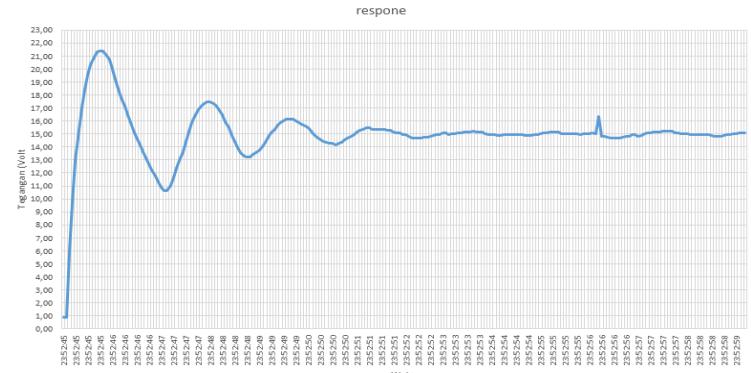
Gambar 4. 22 Grafik dengan nilai $K_p=1$; $K_i=0,1$; $K_d=0,5$

4. $K_p=1$; $K_i=0,1$; $Kd=0,6$



Gambar 4. 23 Grafik dengan nilai $K_p=1$; $K_i=0,1$; $Kd=0,6$

5. $Kp=1$; $Ki=0,1$; $Kd=0,8$



Gambar 4. 24 Grafik dengan nilai $Kp=1$; $Ki=0,1$; $Kd=0,8$

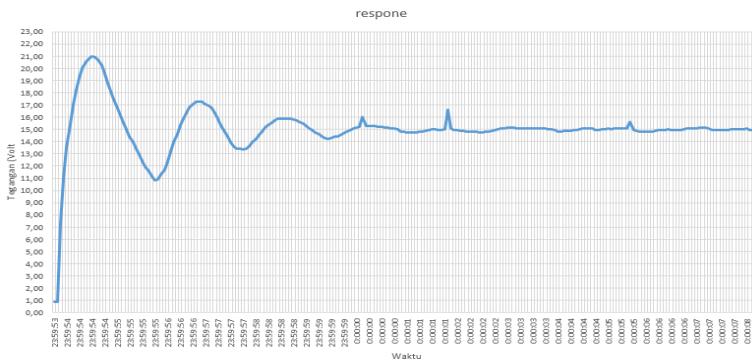
Dari data pengujian Tabel 4.4 diatas dapat disimpulkan bahwa ketika parameter K_p dan K_i tetap dan parameter K_d diubah maka perubahan yang terjadi secara signifikan yaitu pada nilai *overshoot* sedangkan untuk nilai *settling time*, *rise time*, *steady state response*, *transien response* dan *steady state error* hampir memiliki persentase yang sama dan untuk nilai respons yang paling baik ketika parameter kd bernilai 0,1.

Tabel 4. 5 Nilai K_p dan nilai K_d tetap, sedangkan nilai K_i berubah-ubah

No	K _p	K _i	K _d
1	1	0,1	0,1
2	1	0,5	0,1
3	1	1	0,1

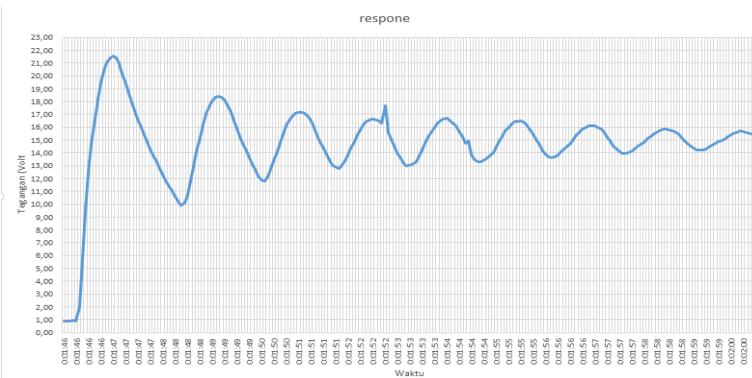
Dari tabel diatas untuk melihat grafik respon dapat dilihat pada Gambar 4.20 sampai dengan Gambar 4.22

1. K_p=1; K_i=0,1;Kd= 0,1



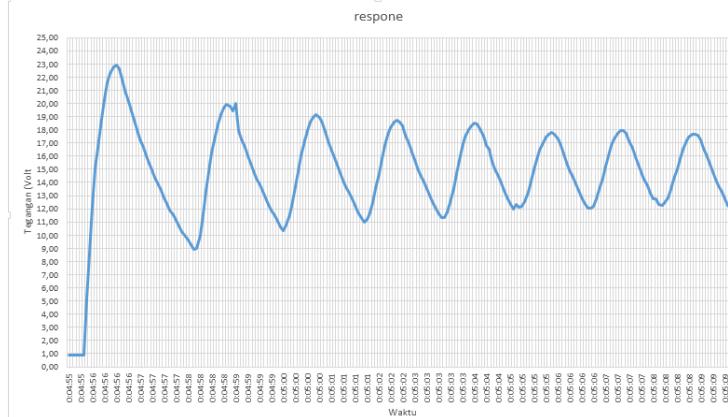
Gambar 4. 25 Grafik dengan nilai K_p=1; K_i=0,1;Kd= 0,1

2. K_p=1; K_i=0,5;Kd= 0,1;



Gambar 4. 26 Grafik dengan nilai K_p=1; K_i=0,5;Kd= 0,1;

3. $K_p=1$; $K_i=1$; $K_d=0,1$;



Gambar 4. 27 Grafik dengan nilai $K_p=1$; $K_i=1$; $K_d=0,1$;

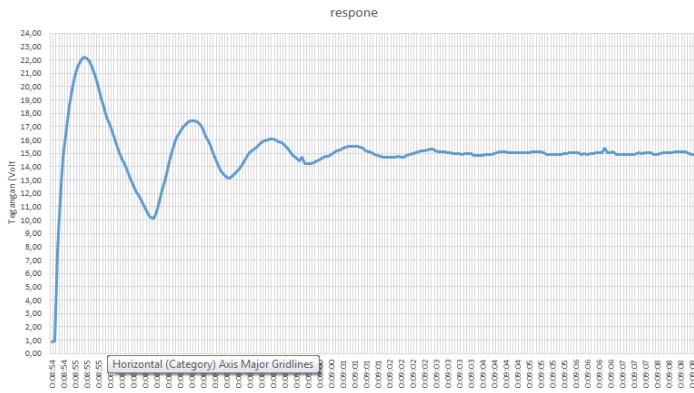
Dari data percobaan dari Tabel 4.5 diatas dapat disimpulkan bahwa sistem mengalami osilasi dengan persentase error yang besar dan untuk beberapa sample data yang diambil respon tidak mengalami kestabilan sehingga untuk menentukan nilai *settling time*, *rise time*, *steady state response*, *transien response* dan *steady state error* cukup sulit. Untuk nilai respon paling baik dengan nilai K_p dan K_d tetap sedangkan untuk nilai K_i tetap yaitu ketika parameter K_i 0,1.

Tabel 4. 6 Nilai K_p yang tetap sedangkan nilai K_i dan nilai K_d berubah-ubah

No	K_p	K_i	K_d
1	1	0,1	1
2	1	0,3	3
3	1	0,5	5
4	1	0,7	7
5	1	1	10

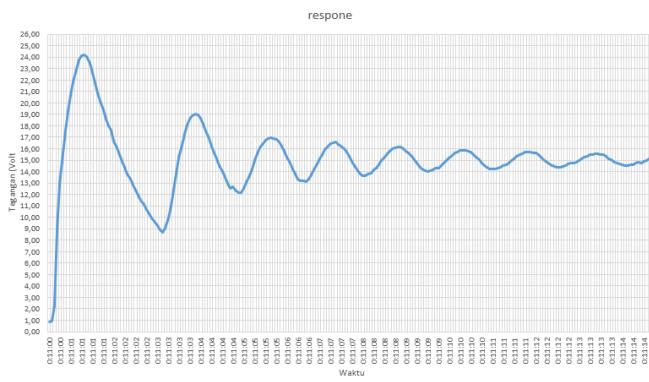
Dari tabel diatas untuk melihat grafik respon dapat dilihat pada Gambar 4.23 sampai dengan Gambar 4.26

1. $K_p=1$; $K_i=0,1$; $K_d=1$



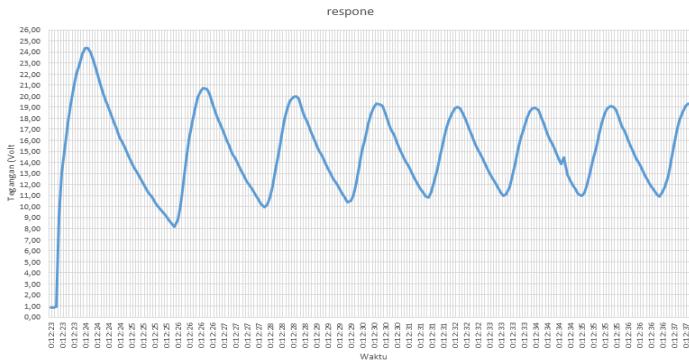
Gambar 4. 28 Grafik dengan nilai $K_p=1$; $K_i=0,1$; $K_d=1$

2. $K_p=1$; $K_i=0,3$; $K_d=3$



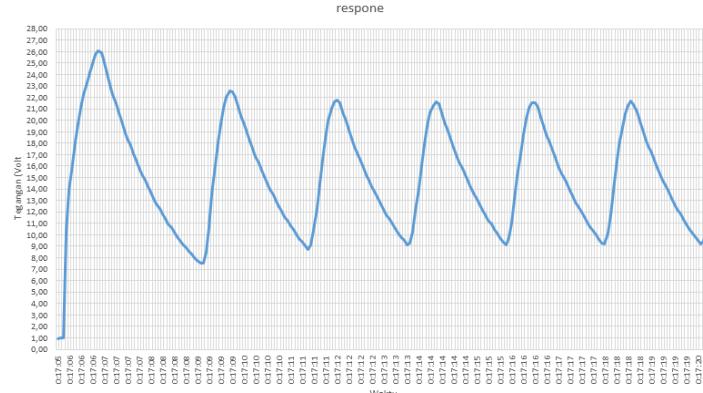
Gambar 4. 29 Grafik dengan nilai $K_p=1$; $K_i=0,3$; $K_d=3$

3. $K_p=1$; $K_i=0,7$; $K_d=7$



Gambar 4. 30 Grafik dengan nilai $K_p=1$; $K_i=0,7$; $K_d=7$

4. $K_p=1$; $K_i=1$; $K_d=10$



Gambar 4. 31 Grafik dengan nilai $K_p=1$; $K_i=1$; $Kd=10$

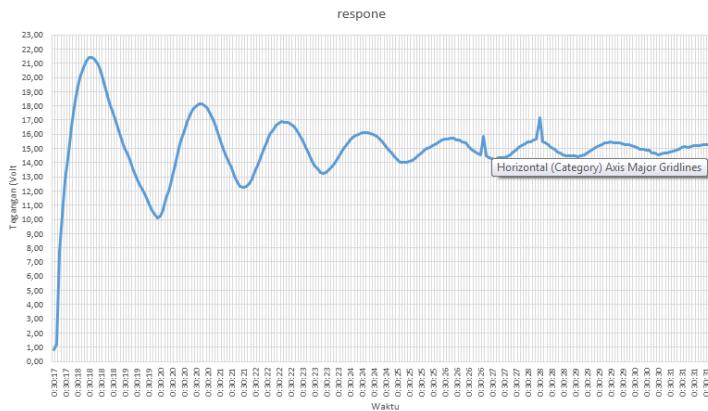
Dari data percobaan Tabel 4.6 diatas dapat disimpulkan bahwa ketika parameter K_p tetap sedangkan parameter K_i dan Kd diubah maka respons akan mengalami osilasi ketika nilai Ki dan Kd dinaikan maka nilai *overshoot*, *settling time*, *rise time*, *steady state response*, *transien response* dan *steady state error* akan semakin besar sehingga nilai respon yang paling baik yaitu ketika parameter $Ki=0,1$ dan $Kd=0,1$.

Tabel 4. 7 Nilai K_p berubah-ubah sedangkan nilai Ki dan Kd tetap

No	K_p	Ki	Kd
1	0,1	0,1	0,1
2	0,4	0,1	0,1
3	0,8	0,1	0,1
4	1	0,1	0,1

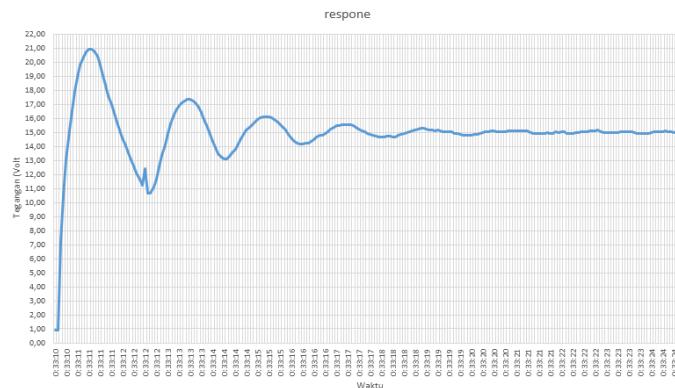
Dari tabel diatas untuk melihat grafik respon dapat dilihat pada Gambar 4.27 sampai dengan Gambar 4.30

1. $K_p=0,4$; $K_i=0,1$; $K_d=0,1$



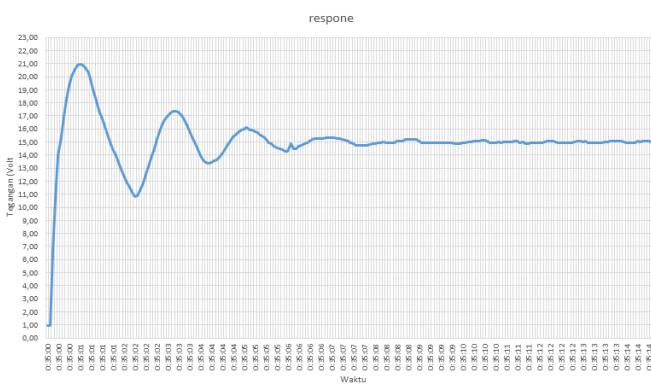
Gambar 4. 32 Grafik dengan nilai $K_p=0,4$; $K_i=0,1$; $K_d=0,1$;

2. $K_p=0,8$; $K_i=0,1$; $K_d=0,1$;



Gambar 4. 33 Grafik dengan nilai $K_p=0,8$; $K_i=0,1$; $K_d=0,1$

3. $K_p=1$; $K_i=0,1$; $K_d=0,1$;



Gambar 4. 34 Grafik dengan nilai $K_p=1$; $K_i=0,1$; $K_d=0,1$;

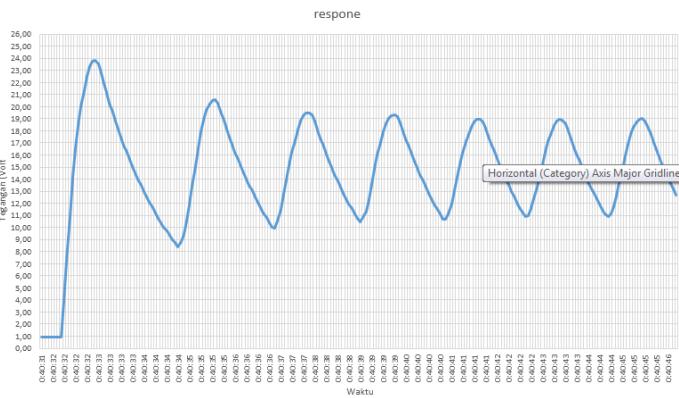
Dari data percobaan diatas dapat disimpulkan bahwa ketika parameter K_i dan K_d tetap sedangkan parameter K_p diubah maka respons akan mengalami osilasi ketika nilai K_p lebih kecil dari 0,1 dan lebih besar 0,1 yang menyebabkan nilai persentase *overshoot* semakin besar karena nilai *overshoot* sistem paling besar dipengaruhi oleh parameter K_p sehingga untuk mendapatkan parameter yang sesuai maka parameter $K_p=1, K_i=0,1$ dan $K_d=0,1$.

Tabel 4. 8 Nilai K_p dan K_d berubah-ubah sedangkan nilai K_i tetap

No	K_p	K_i	K_d
1	0,1	0,7	0,1
2	0,4	0,7	0,4
3	0,8	0,7	0,8
4	1	0,7	1

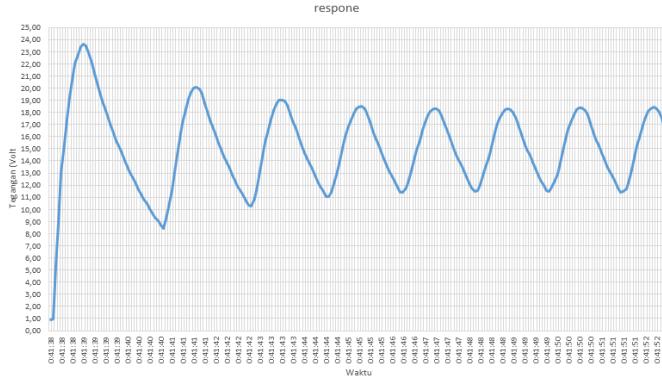
Dari tabel diatas untuk melihat grafik respon dapat dilihat pada Gambar 4.31 sampai dengan Gambar 4.3

1. $K_p=0,1; K_i=0,7; K_d=0,1$



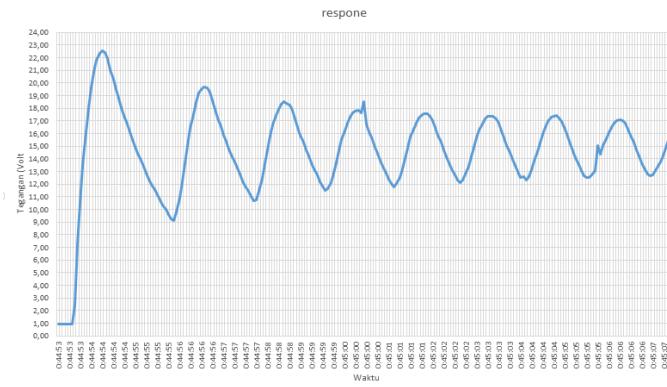
Gambar 4. 35 Grafik dengan nilai $K_p=0,1; K_i=0,7; K_d=0,1$

2. $K_p=0,4$; $K_i=0,7$; $K_d=0,4$



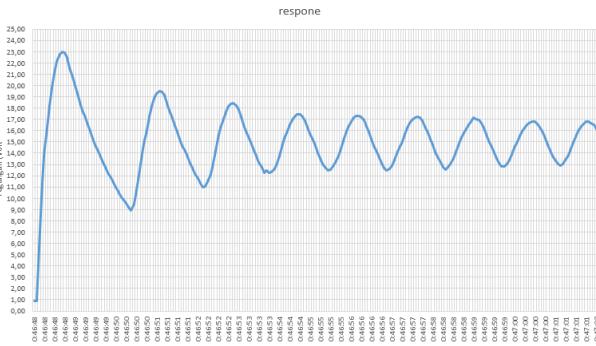
Gambar 4. 36 Grafik dengan nilai $K_p=0,4$; $K_i=0,7$; $K_d=0,4$

3. $K_p=0,8$; $K_i=0,7$; $Kd=0,8$;



Gambar 4. 37 Grafik dengan nilai $K_p=0,8$; $K_i=0,7$; $K_d=0,8$

4. $K_p=1$; $Ki=0,7$; $Kd=1$



Gambar 4. 38 Grafik dengan nilai $K_p=1$; $Ki=0,7$; $Kd=1$

Dari data percobaan diatas nilai osilasi semakin besar ketika parameter K_p diperkecil sehingga nilai *steady state response* sistem akan semakin besar atau bahkan tidak menemui titik stabil dikarenakan nilai dari masing-masing konstanta

PID yang dihasilkan sistem tidak bisa di konversi arduino yang memiliki pwm 0-255. Sehingga dari datra dapat disimpulkan bahwa respon terbaik ketika paramater $K_p=1$, $K_i=0,1$ dan $K_d=0,1$.

BAB V

KESIMPULAN

5.1 Kesimpulan

1. Alat ini mampu menurunkan tegangan hingga 0V dari tegangan *input* (V_{in}). Serta menaikkan tegangan hingga 28V dari tegangan *input* (V_{in}).
2. Persentase *error* untuk *buck converter* yaitu 3,8% dan untuk *boost converter* yaitu 0.95%.
3. Dengan kontrol PID *error* dari kerja sistem dapat direduksi maka sistem yang tidak linear dapat dilinearisasikan.
4. Alat ini mampu menstabilkan tegangan sumber yang tidak stabil atau berubah-ubah.

5.2 Saran

1. Kepresisionan sensor tegangan merupakan hal terpenting untuk sistem *close loop* sehingga nilai *error* dapat diminimalisir dan tegangan keluaran (V_{out}) dari rangkaian *buck-boost* sama dengan *set point*.
2. Karena nilai pwm untuk arduino 0-255 maka untuk menghasilkan tegangan keluaran dengan tingkat ketelitian 0,1 Volt tidak bisa ditambah nilai *error* pada *range* tegangan tertentu lebih besar dari 0,1 Volt.
3. Arduino tidak bisa menghasilkan frekuensi yang pas untuk IC XL6009 karena arduino menyediakan fasilitas frekuensi dengan nilai yang telah ditentukan.

DAFTAR PUSTAKA

- [1] F. Padhillah, Syahrial and S. Saodah, *Perancangan dan Realisasi Konverter DC-DC Tipe Boost Berbasis Mikrokontroller ATMEGA 8535*, vol. 2, p. 2, 2014.
- [2] R. J. Hanson, Implementing Closed Loop Control Algorithms for DC to DC Converters and Arcp Inverters Using the Universial Controller, Amerika Serikat, 1997.
- [3] T. L. Langlois, The analysis of interconnect, High-Power DC-DC Converters for DC Zonal Electrical Distribution, Amerika Serikat, 1997.
- [4] R. Zrafi, S. Ghedira and K. Besbes, "A Bond Graph Approach for the Modeling and Simulation of a *Buck converter*," *Low Power Electronics and Applications*, vol. 8, no. 2, p. 3, 2018.
- [5] M. Jamlay and W. Muhammad, *DUAL FEEDBACK CONTROL DC-DC BOOST CONVERTER*, vol. 04, pp. 91-97, 2014.
- [6] A. Fathurachman, A. Najmurrokhman and Kusnandar, "Perancangan Boost Converter untuk Sistem Pembangkit Listrik Tenagan Surya".
- [7] A. Komarudin, *Desain dan Analisis Proporsional Kontrol Buck-Boost Converter pada Sistem Photovolik*, vol. 12, no. 02, pp. 81-82, 2014.
- [8] S. Arifin and A. Fathoni, "Pemanfaatan Pulse Width Modulation untuk Mengontrol Motor (Studi Kasus Robot Otomatis Dua Deviana)," *Jurnal Ilmiah Teknologi dan Informasi ASIA*, vol. 8, no. 02, pp. 69-70, 2014.
- [9] D. Wisnu, A. Wahyuji and H. Nurhadi, "Perancangan Sistem Kontrol PID untuk Pengendali Sumbu Azimuth Turret pada Turret-gun Kaliber 20mm," vol. 05, p. 01, 2016.
- [10] I. W. R. Ardana, *Simulasi Sistem Kontroler PID Untuk Motor Induksi Menggunakan Perangkat Lunak Matlab/ Simulink*, vol. 7, p. 68, 2008.

LAMPIRAN 1

(Riwayat Hidup Perorangan)

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Ela Sari
Tempat, Tanggal Lahir : Sungailiat, 17 Juni 1998
Alamat : Jl. Sri Menanti No.20
Hp. : 085267855636
Status : Mahasiswi



2. Riwayat Pendidikan

SD Negeri 3 Sungailiat	Lulus 2010
SMP Negeri 1 Sungailiat	Lulus 2013
SMA Negeri 1 Sungailiat	Lulus 2016
DIII POLMAN NEGERI BABEL	2016-Sekarang

3. Pengalaman Kerja

❖ Praktik Kerja Lapangan di PT. Banshu Electric Indonesia	13 Agustus 2018 – 28 Januari 2019
---	--------------------------------------

4. Pengetahuan Bahasa : Bahasa Indonesia dan Bahasa Inggris

5. Hobi : Menari

Sungailiat, 28 Agustus 2019

Ela Sari

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Fahmi Hidayat
Tempat & Tanggal Lahir : Bukit Jaya, 22 Desember 1998
Alamat : Dsn. Bukit Jaya, Ds. Buding
Hp. : 085814247601
Status : Mahasiswa



2. Riwayat Pendidikan

SD Negeri 11 Kelapa Kampit	Lulus 2010
SMP Negeri 2 Kelapa Kampit	Lulus 2013
SMA Negeri 1 Kelapa Kampit	Lulus 2016
DIII POLMAN NEGERI BABEL	2016-Sekarang

3. Pengalaman Kerja

❖ Praktik Kerja Lapangan di PT. Hanabe Kharisma Sejati	10 September – 10 Januari 2019
--	--------------------------------

4. Pengetahuan Bahasa : Bahasa Indonesia

5. Hobi : Sepak bola, Renang, Musik

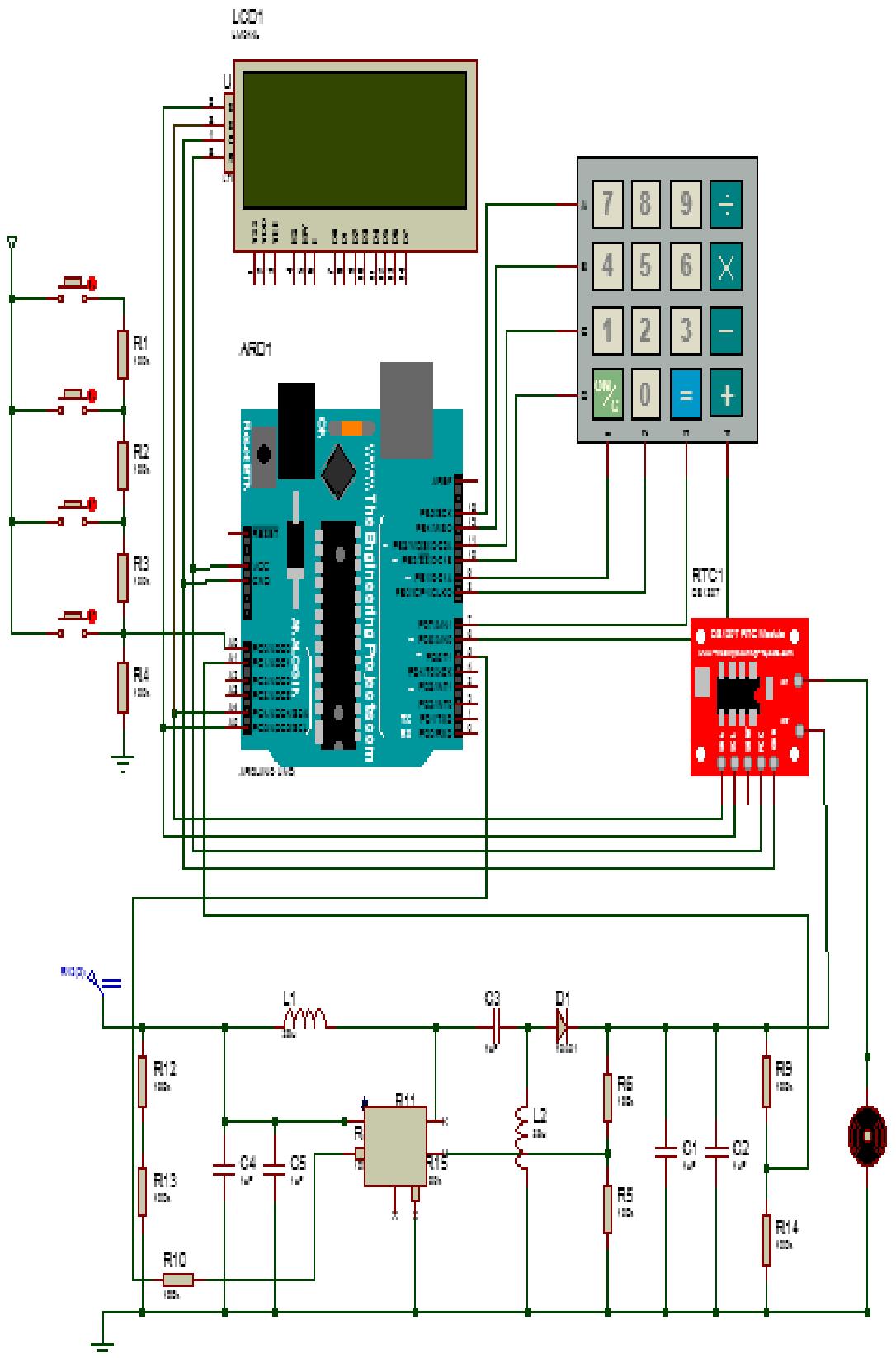
Sungailiat, 28 Agustus 2019



Fahmi Hidayat

LAMPIRAN 2

(*Schematic Rangkaian Kontrol Keseluruhan*)



LAMPIRAN 3

**(*Data Sheet* DC to DC Converter XL
SEMI 6009)**

LAMPIRAN 4

(*Data Sheet* Sensor INA219)

LAMPIRAN 5

(Program Arduino UNO)

```

#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
#include <EEPROM.h>
#include <SoftwareSerial.h>
#include <boarddefs.h>
#define sensor_out A1
#define pwm 6
#define BINTANG 42
#define PAGAR 35
#define NOL 48
#define A 65
#define B 66
#define C 67
#define D 68
#define ROWS 4 //four rows
#define COLS 4
#define ADDR_BACKLIGHT 0
#define ADDR_ARAH 1
#define BACKLIGHT_ON HIGH
#define BACKLIGHT_OFF !BACKLIGHT_ON
#define LCD_COL 20
#define LCD_ROW 4
#define pb_up 3
#define pb_down 2
#define ok 1000
#define back 200
#include <Adafruit_INA219.h>
String str;
const int INA_addr = 0x40; // INA219 address

Adafruit_INA219 ina219(INA_addr);

float voltase = 00;
float arus = 00; // diukur menggunakan MiliAmpere
float daya = 00;

float RawValue= 0;
float Current = 0;
char customKey;
float teg,respon;
int Panjang=0;//deklarasi variable pada matlab
float matlab_data;//deklarasi variable pada matlab
char input[10];
float kp,ki,kd,pid,integral,proporsional,derivatif,error,kesalahan;

```

```

float data,nilaikp,nilaiki,nilaikd,sp,nilaisp,sensor1;
byte pb[]={pb_up,pb_down,ok,back};

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
boolean backlight = BACKLIGHT_ON;
int tegangan,ref,up1=0,program;
byte tekan=0;
float tegangan22 = 00;

uint16_t icr = 0xffff;
String line;

int Data=0;
int result1,result2,result3,a,b,c,d,keydd1,keydd2,keydd3;
int langkah=0;
char keys[ROWS][COLS] =
{
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {13,12,11,10}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {9,8,7,5};
Keypad customKeypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS,
COLS );
// Set the LCD address to 0x27 for a 16 chars and 2 line display
//LiquidCrystal_I2C lcd(0x27, 16, 2);
byte pin=7;
long stepy,hasil,hasil1,hasil2,hasil3,hasil4,hasil5,data_ir1,data_ir2,data_ir3;

String Menu_Mode[2]=
{
    "1. default",
    "2. set PID"
};
String Menu_acces[2]=
{
    "1. Keypad",
    "2. nodeMcu"
};
String Menu_Metode[2]=
{
    "1. parallax",
    "2. Matlab"
};

```

```

void setup()
{
    Serial.begin(9600);
    ina219.begin();
    while(!Serial);
    lcd.begin(LCD_COL,LCD_ROW);
    //pinMode(max471In, INPUT);
    Serial.println("CLEARSHHEET"); // clears sheet starting at row 1
    Serial.println("CLEARDATA"); // clears sheet starting at row 1
    Serial.println("LABEL,Timer,tegangan");
    lcd.setBacklight(backlight);
    pinMode(pwm,OUTPUT);
    for(a=0;a<=3;a++)
    {
        digitalWrite(pb[a],INPUT_PULLUP);
        digitalWrite(ok,INPUT_PULLUP);
        digitalWrite(back,INPUT_PULLUP);
    }
}
void loop()
{
    while(langkah==0)
    {
        analogWrite(pwm,0);
        data=analogRead(sensor_out);
        teg =(data/343.82)*10;
        lcd.setCursor(15,2);
        lcd.print(teg);
        up=constrain(up,0,1);
        lcd.setCursor(0,0);
        lcd.print(" Choose acces ");
        lcd.setCursor(0,1);
        lcd.print(" ===== ");
        lcd.setCursor(0,2);
        lcd.print(Menu_acces[up]);
        if(analogRead(A3)>=200 && analogRead(A3)<=300){up++;lcd.clear();}
        if(analogRead(A3)>=301 && analogRead(A3)<=400){up--;lcd.clear();}

        if((analogRead(A3)>=401&&analogRead(A3)<=800))&&up==1){langkah=99
9;lcd.clear();}

        if((analogRead(A3)>=401&&analogRead(A3)<=800))&&up==0){langkah=99
9;lcd.clear();}
    }
    while(langkah==999)
}

```

```

{
analogWrite(pwm,0);
if(analogRead(A3)>=200&&analogRead(A3)<=300){up1++;lcd.clear();}
if(analogRead(A3)>=301&&analogRead(A3)<=409){up1--;lcd.clear();}
if(analogRead(A3)>=470&&up1==0){langkah=1000;lcd.clear();up1=0;c=1;}
if(analogRead(A3)>=470&&up1==1){langkah=1000;lcd.clear();up1=0;c=2;}
up1=constrain(up1,0,1);
lcd.setCursor(0,0);
lcd.print(" Choose Metode");
lcd.setCursor(0,1);
lcd.print(" ======");
lcd.setCursor(0,2);
lcd.print(Menu_Metode[up1]);
}
While(langkah==1000)
{
analogWrite(pwm,0);
if(analogRead(A3)>=200&&analogRead(A3)<=300)up1++;
if(analogRead(A3)>=301&&analogRead(A3)<=409)up1--;
if(analogRead(A3)>=470&&up1==0){langkah=1;program=111;lcd.clear();}
if(analogRead(A3)>=470&&up1==1){langkah=1;program=1;lcd.clear();}
Serial.println("//////////");
up1=constrain(up1,0,1);
lcd.setCursor(0,0);
lcd.print(" Choose Mode ");
lcd.setCursor(0,1);
lcd.print(" ======");
lcd.setCursor(0,2);
lcd.print(Menu_Mode[up1]);
}
while(langkah==1)
{
while(program==111)
{
if(d==1)c=1;
if(d==2)c=2;
b=1;
analogWrite(pwm,0);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Set Parameter");
lcd.setCursor(0,1);
lcd.print(" =====");
lcd.setCursor(0,2);
lcd.print("sp= ");
up1=0;
}
}

```

```

kp=1;
ki=0.1;
kd=0.1;
setspl();
cekpid();
}
while(program==1)
{
if(d==1)c=1;
if(d==2)c=2;
b=2;
analogWrite(pwm,0);
lcd.clear();
analogWrite(pwm,10);
data=analogRead(sensor_out);
teg =(data/343.82)*10;
analogWrite(pwm,0);
lcd.setCursor(14,0);
lcd.print(teg);
lcd.setCursor(0,0);
lcd.print("Set Parameter");
lcd.setCursor(0,1);
lcd.print(" ======");
lcd.setCursor(0,2);
lcd.print("Kp= ");
lcd.setCursor(0,3);
lcd.print("Ki= ");
lcd.setCursor(10,2);
lcd.print("Kd= ");
lcd.setCursor(10,3);
lcd.print("Sp= ");
switch(customKey)
{
case '0' ... '9':
break;

case '#':
break;

case '*':
break;

case 'A':
break;

case 'B':

```

```

break;

case 'C':
break;

case 'D':
break;
}
setkp();
setki();
setkd();
setsps();
cekpid();
}
while(program==3)
{
analogWrite(pwm,0);
lcd.setCursor(3,0);
lcd.print("Set Parameter");
lcd.setCursor(0,1);
lcd.print(" ======");
lcd.setCursor(0,2);
lcd.print("Kp= ");
lcd.setCursor(0,3);
lcd.print("Ki= ");
lcd.setCursor(10,2);
lcd.print("Kd= ");
lcd.setCursor(10,3);
lcd.print("Sp= ");
}
///////////////////////////////PROGRAM SENSOR INA219///////////////////////
while(program==10)
{
while(c==1)
{
voltase = ina219.getBusVoltage_V(); //command untuk pembacaan tegangan
arus = ina219.getCurrent_mA(); //command untuk pembacaan arus
daya = tegangan * (arus/1000);
tegangan=keydd3;

if(tegangan>=1 && tegangan<10)
{
data=analogRead(sensor_out);
teg =(data/962.64)*28;
}
if(tegangan>=10 && tegangan<12)

```

```

{
data=analogRead(sensor_out);
teg =(data/727.44)*28;
}
if(tegangan>=12&&tegangan<14)
{
data=analogRead(sensor_out);
teg =(data/713.132)*28;
}
if(tegangan>=15&&tegangan<=17)
{
data=analogRead(sensor_out);
teg =(data/699.44)*28;
}
if(tegangan>=18&&tegangan<=19)
{
data=analogRead(sensor_out);
teg =(data/691.32)*28;
}
if(tegangan>=20&&tegangan<=28)
{
data=analogRead(sensor_out);
teg =(data/683.754)*28;
}
long keyd=customKeypad .getKey();
Serial.print((String)"DATA,TIME,");
Serial.println(teg);
lcd.setCursor(0,0);
lcd.print("set point = ");
lcd.setCursor(12,0);
lcd.print(tegangan);
lcd.setCursor(0,1);
lcd.print("Vout = ");
lcd.setCursor(7,1);
lcd.print(teg);lcd.print(" Volt");
lcd.setCursor(0,2);
lcd.print("arus = ");lcd.print(arus); lcd.print("mA");
lcd.setCursor(0,3);
lcd.print("daya = ");lcd.print(arus/1000*teg);lcd.print(" watt ");

respon = hitungPID(tegangan,teg);
respon=constrain(respon,0,255);
analogWrite(pwm,respon);

```

```

if(((analogRead(A3)>=400 && analogRead(A3)<=700))&&
b==1){b=0;langkah=1;program=111;kp=0;ki=0;kd=0;sp=0;keydd3=0;lcd.clea
r();c=0;d=1;}
if(((analogRead(A3)>=400 && analogRead(A3)<=700))&&
b==2){b=0;langkah=1;program=1;kp=0;ki=0;kd=0;sp=0;keydd3=0;lcd.clear()
;c=0;d=1;}

if(analogRead(A3)>=800){langkah=0;program=0;kp=0;ki=0;kd=0;sp=0;keydd
3=0;lcd.clear();c=0;d=0;}

TCCR0B = (TCCR0B & 0b11111000) / 5;
setPwmFrequency(pwm,5);
}

while(c==2)
{
voltase = ina219.getBusVoltage_V(); //command untuk pembacaan tegangan
arus = ina219.getCurrent_mA(); //command untuk pembacaan arus
daya = tegangan * (arus/1000);
tegangan=keydd3;

if(tegangan>=1&&tegangan<10)
{
data=analogRead(sensor_out);
teg =(data/962.64)*28;
matlab_data=962.64;
}
if(tegangan>=10&&tegangan<12)
{
data=analogRead(sensor_out);
teg =(data/727.44)*28;
matlab_data=727.44;
}
if(tegangan>=12&&tegangan<14)
{
data=analogRead(sensor_out);
teg =(data/713.132)*28;
matlab_data=713.132;
}
if(tegangan>=15&&tegangan<=17)
{
data=analogRead(sensor_out);
teg =(data/699.44)*28;
matlab_data=699.44;
}
if(tegangan>=18&&tegangan<=19)

```

```

{
  data=analogRead(sensor_out);
  teg =(data/691.32)*28;
  matlab_data=691.32;
}
  if(tegangan>=20 && tegangan<=28)
{
  data=analogRead(sensor_out);
  teg =(data/683.754)*28;
  matlab_data=63.754;
}
long keyd=customKeypad .getKey();
lcd.setCursor(0,0);
lcd.print("set point = ");
lcd.setCursor(12,0);
lcd.print(tegangan);
lcd.setCursor(0,1);
lcd.print("Vout = ");
lcd.setCursor(7,1);
lcd.print(teg);lcd.print(" Volt");
lcd.setCursor(0,2);
lcd.print("arus = ");lcd.print(arus); lcd.print("mA");
lcd.setCursor(0,3);
lcd.print("daya = ");lcd.print(arus/1000*teg);lcd.print(" watt ");

respon = hitungPID(tegangan,teg);
respon=constrain(respon,0,255);
analogWrite(pwm,respon);

if((analogRead(A3)>=400 && analogRead(A3)<=700))&&
b==1){b=0;langkah=1;program=111;kp=0;ki=0;kd=0;sp=0;keydd3=0;lcd.clear();c=0;d=2;}
  if((analogRead(A3)>=400 && analogRead(A3)<=700))&&
b==2){b=0;langkah=1;program=1;kp=0;ki=0;kd=0;sp=0;keydd3=0;lcd.clear()
;c=0;d=2;}

if(analogRead(A3)>=800){langkah=0;program=0;kp=0;ki=0;kd=0;sp=0;keydd
3=0;lcd.clear();c=0;d=0;}

TCCR0B = (TCCR0B & 0b11111000) / 5;
setPwmFrequency(pwm,5);
//////////PROGRAM MENAMPILKAN GELOMBANG PADA MATLAB/////////
  if((input[0]=Serial.read())>0)
{
  Panjang=Serial.available()+1;
  for(int i=1; i<Panjang; i++)

```

```

{
    input[i]=Serial.read();
}
input[Panjang] = "\0";
Data=atoi(input);
if(Data==1)
{
    float sensor1=analogRead(sensor_out);
    teg=(sensor1/matlab_data)*28;
    Serial.println(teg);
}
}
}}}

///////////////////////////////PROGRAM UNTUK MENGATUR FREKUENSI///////////////////
void setPwmFrequency(int pin, int divisor)
{
byte mode;
if(pin == 5 || pin == 6 || pin == 9 || pin == 10) {
switch(divisor) {
case 1: mode = 0x01; break;
case 8: mode = 0x02; break;
case 64: mode = 0x03; break;
case 256: mode = 0x04; break;
case 1024: mode = 0x05; break;
default: return;
}
if(pin == 5 || pin == 6)
{
    TCCR0B = TCCR0B & 0b11111000 / mode;
}
else
{
    TCCR1B = TCCR1B & 0b11111000 / mode;
}
}
else if(pin == 3 || pin == 11)
{
switch(divisor)
{
case 1: mode = 0x01; break;
case 8: mode = 0x02; break;
case 32: mode = 0x03; break;
case 64: mode = 0x04; break;
case 128: mode = 0x05; break;
}
}
}

```

```

        case 256: mode = 0x06; break;
        case 1024: mode = 0x07; break;
        default: return;
    }
    TCCR2B = TCCR2B & 0b11111000 / mode;
}
}

///////////////////////////////PROGRAM PID///////////////////////////
float hitungPID(float tegangan,float teg)
{
    kesalahan=tegangan-teg;
    proporsional=kesalahan*kp;
    integral=integral+(error*ki);
    derivatif=(error-kesalahan)*kd;
    integral+=kesalahan;
    error=kesalahan;
    pid=(proporsional+integral+derivatif)/0.3;
    return (pid);
}
void setkp()
{
    sensor1=analogRead(sensor_out);
    teg=(sensor1/631.28)*28;
    lcd.setCursor(14,0);
    lcd.print(teg);
    analogWrite(pwm,0);
    lcd.setCursor(3,2);
    lcd.print(kp);
    sensor1=analogRead(sensor_out);
    teg=(sensor1/631.28)*28;
    analogWrite(pwm,0);
    lcd.setCursor(14,0);
    lcd.print(teg);
    customKey = customKeypad.getKey();

    if((customKey >= '0' && customKey <= '9'))
    {
        kp = kp * 10 + (customKey-'0');
    }
    if(customKey == 'A'){
        kp = kp/10;
        nilaikp = kp;
    }

    if(customKey == 'B'){

```

```

kp = kp/100;
nilaikp = kp;
}
if(customKey == 'C'){
kp = kp/1000;
nilaikd = kd;
}
if(customKey == '*'){
nilaikp = kp;
return;
}
setkp();
}
void setki(){
sensor1=analogRead(sensor_out);
teg=(sensor1/631.28)*28;
lcd.setCursor(14,0);
lcd.print(teg);
customKey = customKeypad.getKey();
lcd.setCursor(3,3);
lcd.print(ki);
if(customKey >= '0' && customKey <= '9')
{
ki = ki * 10 + (customKey - '0');
}
if(customKey == 'A'){
ki = ki/10;
nilaiki = ki;
}

if(customKey == 'B'){
ki = ki/100;
nilaiki = ki;
}
if(customKey == 'C'){
ki = ki/1000;
nilaikd = kd;
}

if(customKey == '*'){
nilaiki = ki;
return;
}

setki();
}

```

```

void setkd()
{
    sensor1=analogRead(sensor_out);
    teg=(sensor1/631.28)*28;
    lcd.setCursor(14,0);
    lcd.print(teg);
    customKey = customKeypad.getKey();
    lcd.setCursor(13,2);
    lcd.print(kd);
    if(customKey >= '0' && customKey <= '9')
    {
        kd = kd * 10 + (customKey - '0');
    }

    if(customKey == 'A'){
        kd = kd/10;
        nilaikd = kd;
    }

    if(customKey == 'B'){
        kd = kd/100;
        nilaikd = kd;
    }

    if(customKey == 'C'){
        kd = kd/1000;
        nilaikd = kd;
    }

    if(customKey == '*'){
        nilaikd = kd;
        return;
    }

    setkd();
}

void setsp(){
    sensor1=analogRead(sensor_out);
    teg=(sensor1/631.28)*28;
    lcd.setCursor(14,0);
    lcd.print(teg);
    customKey = customKeypad.getKey();
    lcd.setCursor(13,3);
    lcd.print(sp);
    if(customKey >= '0' && customKey <= '9')
    {

```

```

sp = sp * 10 + (customKey - '0');
lcd.setCursor(0,1);
}

if(customKey == '*'){
    keydd3 = sp;
    lcd.clear();
    return;
}
setsp();
}
void setsp1(){
    sensor1=analogRead(sensor_out);
    teg=(sensor1/631.28)*28;
    lcd.setCursor(14,0);
    lcd.print(teg);
    customKey = customKeypad.getKey();
    lcd.setCursor(4,2);
    lcd.print(sp);
    if(customKey >= '0' && customKey <= '9')
    {
        sp = sp * 10 + (customKey - '0');
        lcd.setCursor(0,1);
    }

    if(customKey == '*'){
        keydd3 = sp;
        lcd.clear();
        return;
    }
}
setsp1();
}
void cekpid()
{
    customKey = customKeypad.getKey();
    lcd.setCursor(0,0);lcd.print("Nilai Kp=");lcd.print(kp);
    lcd.setCursor(0,1);lcd.print("Nilai Ki=");lcd.print(ki);
    lcd.setCursor(0,2);lcd.print("Nilai Kd=");lcd.print(kd);
    lcd.setCursor(0,3);lcd.print("Setpoint=");lcd.print(sp);
    if(customKey == '*'){
        program=10;
        lcd.clear();
        delay(200);
        return;
    }
    cekpid();}
```



Zerø-Drift, Bi-Directional CURRENT/POWER MONITOR with I²C™ Interface

FEATURES

- SENSES BUS VOLTAGES FROM 0V TO +26V
- REPORTS CURRENT, VOLTAGE, AND POWER
- 16 PROGRAMMABLE ADDRESSES
- HIGH ACCURACY: 1% (Max) OVER TEMPERATURE
- FILTERING OPTIONS
- CALIBRATION REGISTERS
- SOT23-8 AND SO-8 PACKAGES

APPLICATIONS

- SERVERS
- TELECOM EQUIPMENT
- NOTEBOOK COMPUTERS
- POWER MANAGEMENT
- BATTERY CHARGERS
- WELDING EQUIPMENT
- POWER SUPPLIES
- TEST EQUIPMENT

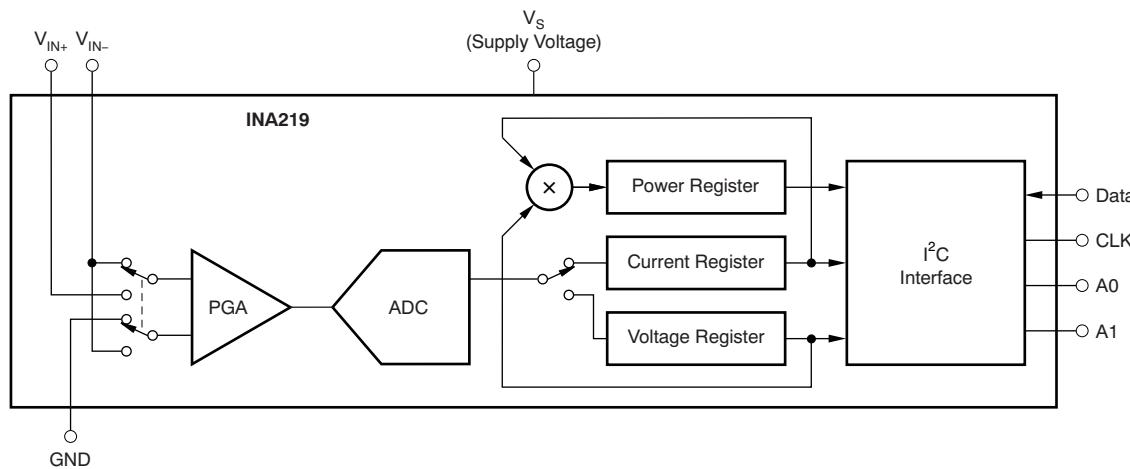
DESCRIPTION

The INA219 is a high-side current shunt and power monitor with an I²C interface. The INA219 monitors both shunt drop and supply voltage, with programmable conversion times and filtering. A programmable calibration value, combined with an internal multiplier, enables direct readouts in amperes. An additional multiplying register calculates power in watts. The I²C interface features 16 programmable addresses.

The INA219 senses across shunts on buses that can vary from 0V to 26V. The device uses a single +3V to +5.5V supply, drawing a maximum of 1mA of supply current. The INA219 operates from -40°C to +125°C.

RELATED PRODUCTS

DESCRIPTION	DEVICE
Current/Power Monitor with Watchdog, Peak-Hold, and Fast Comparator Functions	INA209
Zerø-Drift, Low-Cost, Analog Current Shunt Monitor Series in Small Package	INA210 - INA214



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

I²C is a trademark of NXP Semiconductors.

All other trademarks are the property of their respective owners.



This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

ORDERING INFORMATION⁽¹⁾

PRODUCT	PACKAGE-LEAD	PACKAGE DESIGNATOR	PACKAGE MARKING
INA219	SO-8	D	I219A
	SOT23-8	DCN	A219

- (1) For the most current package and ordering information see the Package Option Addendum at the end of this document, or see the TI web site at www.ti.com.

ABSOLUTE MAXIMUM RATINGS⁽¹⁾

Over operating free-air temperature range (unless otherwise noted).

		INA219	UNIT
Supply Voltage, V _S		6	V
Analog Inputs, V _{IN+} , V _{IN-}	Differential (V _{IN+}) – (V _{IN-}) ⁽²⁾	-26 to +26	V
	Common-Mode	-0.3 to +26	V
SDA		GND – 0.3 to +6	V
SCL		GND – 0.3 to V _S + 0.3	V
Input Current Into Any Pin		5	mA
Open-Drain Digital Output Current		10	mA
Operating Temperature		-40 to +125	°C
Storage Temperature		-40 to +150	°C
Junction Temperature		+150	°C
ESD Ratings	Human Body Model	4000	V
	Charged-Device Model	750	V
	Machine Model (MM)	200	V

- (1) Stresses above these ratings may cause permanent damage. Exposure to absolute maximum conditions for extended periods may degrade device reliability. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those specified is not implied.
- (2) V_{IN+} and V_{IN-} may have a differential voltage of -26V to +26V; however, the voltage at these pins must not exceed the range -0.3V to +26V.

ELECTRICAL CHARACTERISTICS: $V_S = +3.3V$

Boldface limits apply over the specified temperature range, $T_A = -25^\circ\text{C}$ to $+85^\circ\text{C}$.

At $T_A = +25^\circ\text{C}$, $V_{IN+} = 12V$, $V_{SENSE} = (V_{IN+} - V_{IN-}) = 32mV$, PGA = $\div 1$, and BRNG⁽¹⁾ = 1, unless otherwise noted.

PARAMETER	TEST CONDITIONS	INA219			UNIT
		MIN	TYP	MAX	
INPUT					
Full-Scale Current Sense (Input) Voltage Range					
	PGA = $\div 1$	0		± 40	mV
	PGA = $\div 2$	0		± 80	mV
	PGA = $\div 4$	0		± 160	mV
	PGA = $\div 8$	0		± 320	mV
Bus Voltage (Input Voltage) Range ⁽²⁾					
	BRNG = 1	0		32	V
	BRNG = 0	0		16	V
Common-Mode Rejection	CMRR				
Offset Voltage, RTI ⁽³⁾	V_{OS}	$V_{IN+} = 0V$ to 26V	100	120	dB
		PGA = $\div 1$	± 10	± 100	μV
		PGA = $\div 2$	± 20	± 125	μV
		PGA = $\div 4$	± 30	± 150	μV
		PGA = $\div 8$	± 40	± 200	μV
vs Temperature				0.1	$\mu\text{V}/^\circ\text{C}$
vs Power Supply	PSRR			10	$\mu\text{V/V}$
Current Sense Gain Error		$V_S = 3V$ to 5.5V		± 40	m%
vs Temperature				10	$\text{ppm}/^\circ\text{C}$
Input Impedance		Active Mode			
				20	μA
				20 320	$\mu\text{A} \parallel k\Omega$
Input Leakage ⁽⁴⁾		Power-Down Mode			
				0.1	μA
				0.1	μA
DC ACCURACY					
ADC Basic Resolution				12	Bits
1 LSB Step Size					
	Shunt Voltage			10	μV
	Bus Voltage			4	mV
Current Measurement Error				± 0.2	%
over Temperature				± 0.5	%
Bus Voltage Measurement Error				± 0.2	%
over Temperature				± 0.5	%
Differential Nonlinearity				± 0.1	LSB
ADC TIMING					
ADC Conversion Time		12-Bit	532	586	μs
		11-Bit	276	304	μs
		10-Bit	148	163	μs
		9-Bit	84	93	μs
Minimum Convert Input Low Time			4		

(1) BRNG is bit 13 of the [Configuration Register](#)

(2) This parameter only expresses the full-scale range of the ADC scaling. In no event should more than 26V be applied to this device.

(3) Referred-to-input (RTI).

(4) Input leakage is positive (current flowing into the pin) for the conditions shown at the top of the table. Negative leakage currents can occur under different input conditions.

ELECTRICAL CHARACTERISTICS: $V_S = +3.3V$ (continued)

Boldface limits apply over the specified temperature range, $T_A = -25^\circ\text{C}$ to $+85^\circ\text{C}$.

At $T_A = +25^\circ\text{C}$, $V_{IN+} = 12\text{V}$, $V_{SENSE} = (V_{IN+} - V_{IN-}) = 32\text{mV}$, PGA = 1, and BRNG = 1, unless otherwise noted.

PARAMETER	TEST CONDITIONS	INA219			UNIT
		MIN	TYP	MAX	
SMBus					
SMBus Timeout ⁽⁵⁾			28	35	ms
DIGITAL INPUTS (SDA as Input, SCL, A0, A1)					
Input Capacitance			3		pF
Leakage Input Current	$0 \leq V_{IN} \leq V_S$		0.1	1	μA
Input Logic Levels:					
V_{IH}		0.7 (V_S)		6	V
V_{IL}		-0.3		0.3 (V_S)	V
Hysteresis		500			mV
OPEN-DRAIN DIGITAL OUTPUTS (SDA)					
Logic '0' Output Level	$I_{SINK} = 3\text{mA}$		0.15	0.4	V
High-Level Output Leakage Current	$V_{OUT} = V_S$		0.1	1	μA
POWER SUPPLY					
Operating Supply Range		+3		+5.5	V
Quiescent Current			0.7	1	mA
Quiescent Current, Power-Down Mode			6	15	μA
Power-On Reset Threshold			2		V
TEMPERATURE RANGE					
Specified Temperature Range		-25		+85	$^\circ\text{C}$
Operating Temperature Range		-40		+125	$^\circ\text{C}$
Thermal Resistance ⁽⁶⁾	θ_{JA}				
SOT23-8			142		$^\circ\text{C}/\text{W}$
SO-8			120		$^\circ\text{C}/\text{W}$

(5) SMBus timeout in the INA219 resets the interface any time SCL or SDA is low for over 28ms.

(6) θ_{JA} value is based on JEDEC low-K board.

PIN CONFIGURATIONS



PIN DESCRIPTIONS: SOT23-8

SOT23-8 (DCN)		DESCRIPTION
PIN NO	NAME	
1	V _{IN+}	Positive differential shunt voltage. Connect to positive side of shunt resistor.
2	V _{IN-}	Negative differential shunt voltage. Connect to negative side of shunt resistor. Bus voltage is measured from this pin to ground.
3	GND	Ground.
4	V _S	Power supply, 3V to 5.5V.
5	SCL	Serial bus clock line.
6	SDA	Serial bus data line.
7	A0	Address pin. Table 1 shows pin settings and corresponding addresses.
8	A1	Address pin. Table 1 shows pin settings and corresponding addresses.

PIN DESCRIPTIONS: SO-8

SO-8 (D)		DESCRIPTION
PIN NO	NAME	
1	A1	Address pin. Table 1 shows pin settings and corresponding addresses.
2	A0	Address pin. Table 1 shows pin settings and corresponding addresses.
3	SDA	Serial bus data line.
4	SCL	Serial bus clock line.
5	V _S	Power supply, 3V to 5.5V.
6	GND	Ground.
7	V _{IN-}	Negative differential shunt voltage. Connect to negative side of shunt resistor. Bus voltage is measured from this pin to ground.
8	V _{IN+}	Positive differential shunt voltage. Connect to positive side of shunt resistor.

TYPICAL CHARACTERISTICS

At $T_A = +25^\circ\text{C}$, $V_{\text{IN}+} = 12\text{V}$, $V_{\text{SENSE}} = (V_{\text{IN}+} - V_{\text{IN}-}) = 32\text{mV}$, PGA = $\div 1$, and BRNG = 1, unless otherwise noted.

FREQUENCY RESPONSE

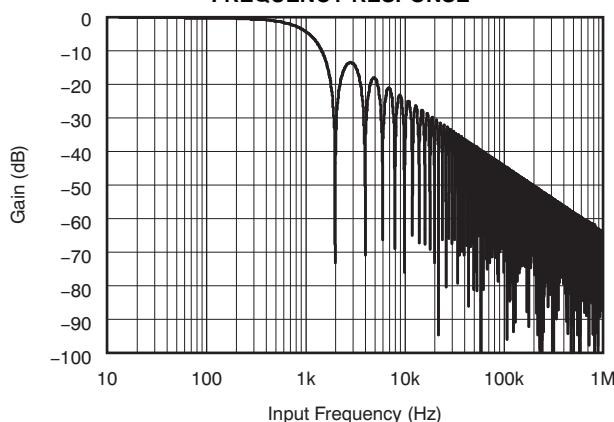


Figure 1.

ADC SHUNT OFFSET vs TEMPERATURE

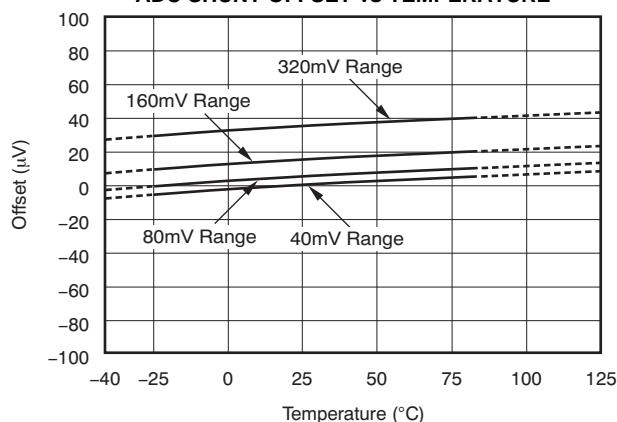


Figure 2.

ADC SHUNT GAIN ERROR vs TEMPERATURE

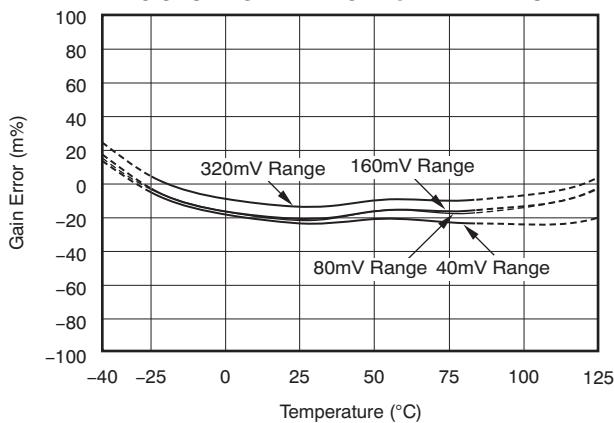


Figure 3.

ADC BUS VOLTAGE OFFSET vs TEMPERATURE

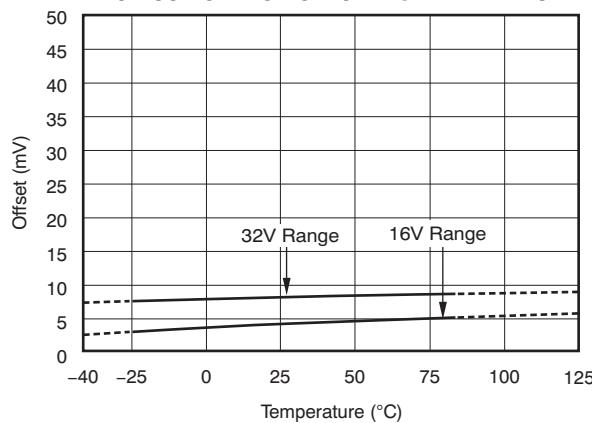


Figure 4.

ADC BUS GAIN ERROR vs TEMPERATURE

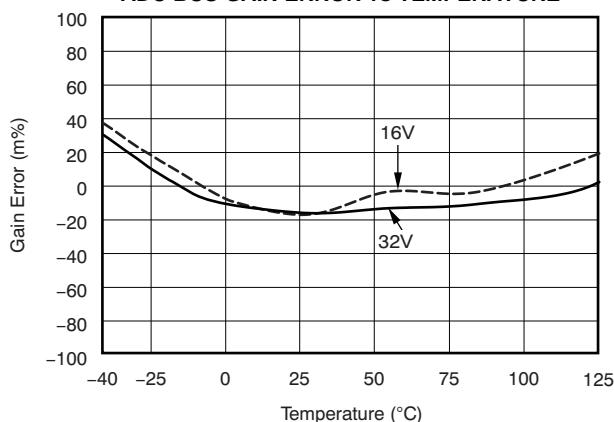


Figure 5.

INTEGRAL NONLINEARITY vs INPUT VOLTAGE

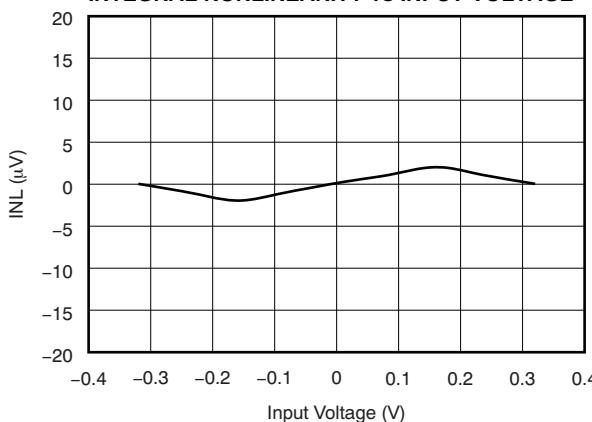


Figure 6.

TYPICAL CHARACTERISTICS (continued)

At $T_A = +25^\circ\text{C}$, $V_{IN+} = 12\text{V}$, $V_{SENSE} = (V_{IN+} - V_{IN-}) = 32\text{mV}$, PGA = $\div 1$, and BRNG = 1, unless otherwise noted.

**INPUT CURRENTS WITH LARGE DIFFERENTIAL VOLTAGES
(V_{IN+} at 12V, Sweep of V_{IN-})**

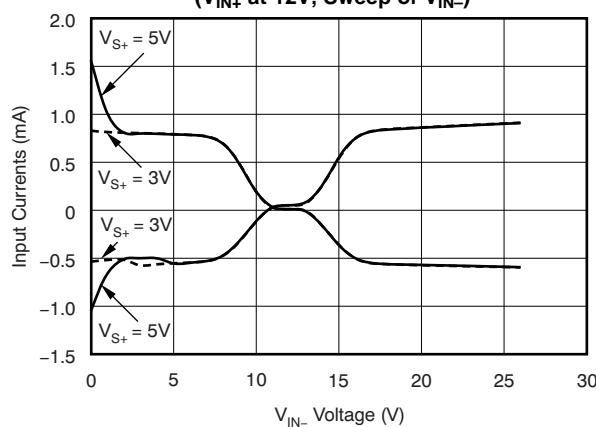


Figure 7.

ACTIVE I_Q vs TEMPERATURE

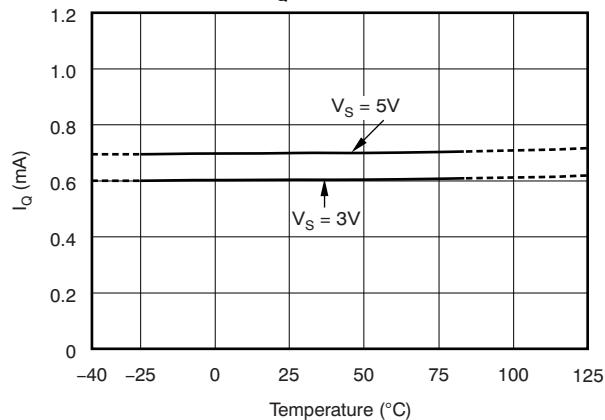


Figure 8.

SHUTDOWN I_Q vs TEMPERATURE

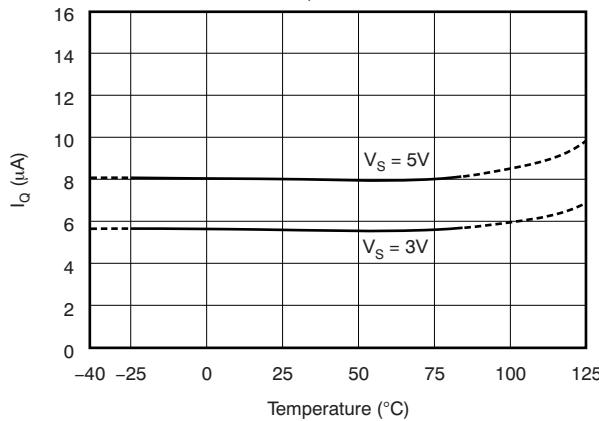


Figure 9.

ACTIVE I_Q vs I²C CLOCK FREQUENCY

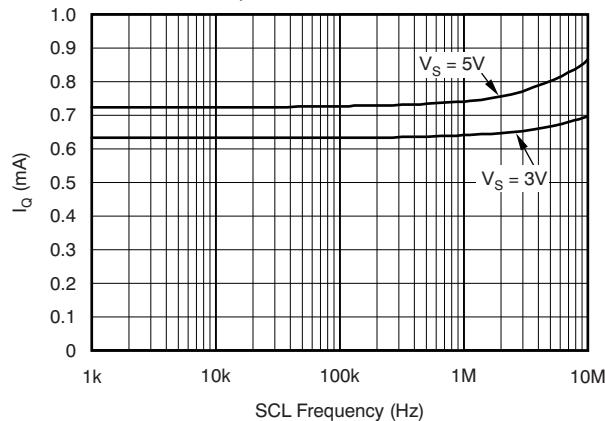


Figure 10.

SHUTDOWN I_Q vs I²C CLOCK FREQUENCY

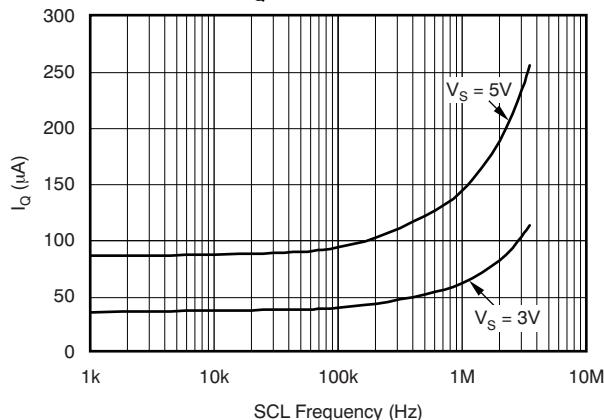


Figure 11.

REGISTER BLOCK DIAGRAM

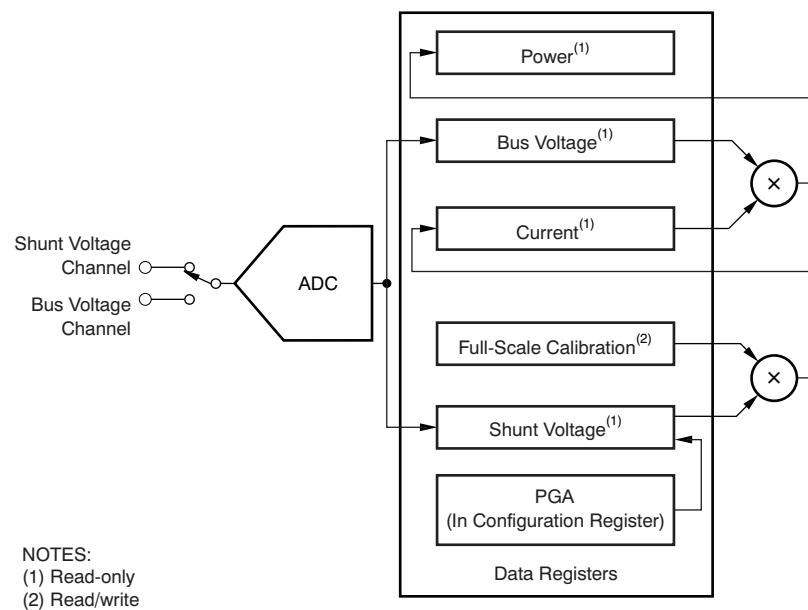


Figure 12. INA219 Register Block Diagram

APPLICATION INFORMATION

The INA219 is a digital current-shunt monitor with an I²C and SMBus-compatible interface. It provides digital current, voltage, and power readings necessary for accurate decision-making in precisely-controlled systems. Programmable registers allow flexible configuration for measurement resolution, and continuous-versus-triggered operation. Detailed register information appears at the end of this data sheet, beginning with [Table 2](#). See the [Register Block Diagram](#) for a block diagram of the INA219.

INA219 TYPICAL APPLICATION

[Figure 13](#) shows a typical application circuit for the INA219. Use a 0.1μF ceramic capacitor for power-supply bypassing, placed as closely as possible to the supply and ground pins.

The input filter circuit consisting of R_{F1} , R_{F2} , and C_F is not necessary in most applications. If the need for filtering is unknown, reserve board space for the components and install 0Ω resistors unless a filter is needed. See the [Filtering and Input Considerations](#) section.

The pull-up resistors shown on the SDA and SCL lines are not needed if there are pull-up resistors on these same lines elsewhere in the system. Resistor values shown are typical: consult either the I²C or SMBus specification to determine the acceptable minimum or maximum values.

BUS OVERVIEW

The INA219 offers compatibility with both I²C and SMBus interfaces. The I²C and SMBus protocols are essentially compatible with one another.

The I²C interface is used throughout this data sheet as the primary example, with SMBus protocol specified only when a difference between the two systems is being addressed. Two bidirectional lines, SCL and SDA, connect the INA219 to the bus. Both SCL and SDA are open-drain connections.

The device that initiates the transfer is called a *master*, and the devices controlled by the master are *slaves*. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates START and STOP conditions.

To address a specific device, the master initiates a START condition by pulling the data signal line (SDA) from a HIGH to a LOW logic level while SCL is HIGH. All slaves on the bus shift in the slave address byte on the rising edge of SCL, with the last bit indicating whether a read or write operation is intended. During the ninth clock pulse, the slave being addressed responds to the master by generating an Acknowledge and pulling SDA LOW.

Data transfer is then initiated and eight bits of data are sent, followed by an Acknowledge bit. During data transfer, SDA must remain stable while SCL is HIGH. Any change in SDA while SCL is HIGH is interpreted as a START or STOP condition.

Once all data have been transferred, the master generates a STOP condition, indicated by pulling SDA from LOW to HIGH while SCL is HIGH. The INA219 includes a 28ms timeout on its interface to prevent locking up an SMBus.

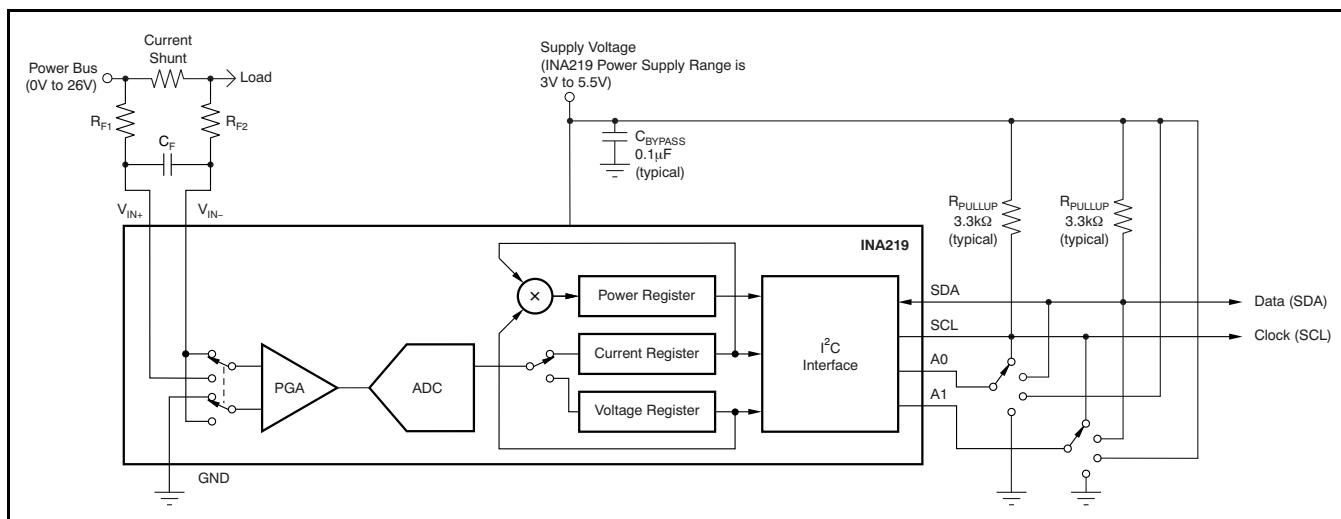


Figure 13. Typical Application Circuit

Serial Bus Address

To communicate with the INA219, the master must first address slave devices via a slave address byte. The slave address byte consists of seven address bits, and a direction bit indicating the intent of executing a read or write operation.

The INA219 has two address pins, A0 and A1. [Table 1](#) describes the pin logic levels for each of the 16 possible addresses. The state of pins A0 and A1 is sampled on every bus communication and should be set before any activity on the interface occurs. The address pins are read at the start of each communication event.

Table 1. INA219 Address Pins and Slave Addresses

A1	A0	SLAVE ADDRESS
GND	GND	1000000
GND	V _{S+}	1000001
GND	SDA	1000010
GND	SCL	1000011
V _{S+}	GND	1000100
V _{S+}	V _{S+}	1000101
V _{S+}	SDA	1000110
V _{S+}	SCL	1000111
SDA	GND	1001000
SDA	V _{S+}	1001001
SDA	SDA	1001010
SDA	SCL	1001011
SCL	GND	1001100
SCL	V _{S+}	1001101
SCL	SDA	1001110
SCL	SCL	1001111

Serial Interface

The INA219 operates only as a slave device on the I²C bus and SMBus. Connections to the bus are made via the open-drain I/O lines SDA and SCL. The SDA and SCL pins feature integrated spike suppression filters and Schmitt triggers to minimize the effects of input spikes and bus noise. The INA219 supports the transmission protocol for fast (1kHz to 400kHz) and high-speed (1kHz to 3.4MHz) modes. All data bytes are transmitted most significant byte first.

WRITING TO/READING FROM THE INA219

Accessing a particular register on the INA219 is accomplished by writing the appropriate value to the register pointer. Refer to [Table 2](#) for a complete list of registers and corresponding addresses. The value for the register pointer as shown in [Figure 17](#) is the first byte transferred after the slave address byte with the R/W bit LOW. Every write operation to the INA219 requires a value for the register pointer.

Writing to a register begins with the first byte transmitted by the master. This byte is the slave address, with the R/W bit LOW. The INA219 then acknowledges receipt of a valid address. The next byte transmitted by the master is the address of the register to which data will be written. This register address value updates the register pointer to the desired register. The next two bytes are written to the register addressed by the register pointer. The INA219 acknowledges receipt of each data byte. The master may terminate data transfer by generating a START or STOP condition.

When reading from the INA219, the last value stored in the register pointer by a write operation determines which register is read during a read operation. To change the register pointer for a read operation, a new value must be written to the register pointer. This write is accomplished by issuing a slave address byte with the R/W bit LOW, followed by the register pointer byte. No additional data are required. The master then generates a START condition and sends the slave address byte with the R/W bit HIGH to initiate the read command. The next byte is transmitted by the slave and is the most significant byte of the register indicated by the register pointer. This byte is followed by an Acknowledge from the master; then the slave transmits the least significant byte. The master acknowledges receipt of the data byte. The master may terminate data transfer by generating a Not-Acknowledge after receiving any data byte, or generating a START or STOP condition. If repeated reads from the same register are desired, it is not necessary to continually send the register pointer bytes; the INA219 retains the register pointer value until it is changed by the next write operation.

[Figure 14](#) and [Figure 15](#) show read and write operation timing diagrams, respectively. Note that register bytes are sent most-significant byte first, followed by the least significant byte. [Figure 16](#) shows the timing diagram for the SMBus Alert response operation. [Figure 17](#) illustrates a typical register pointer configuration.

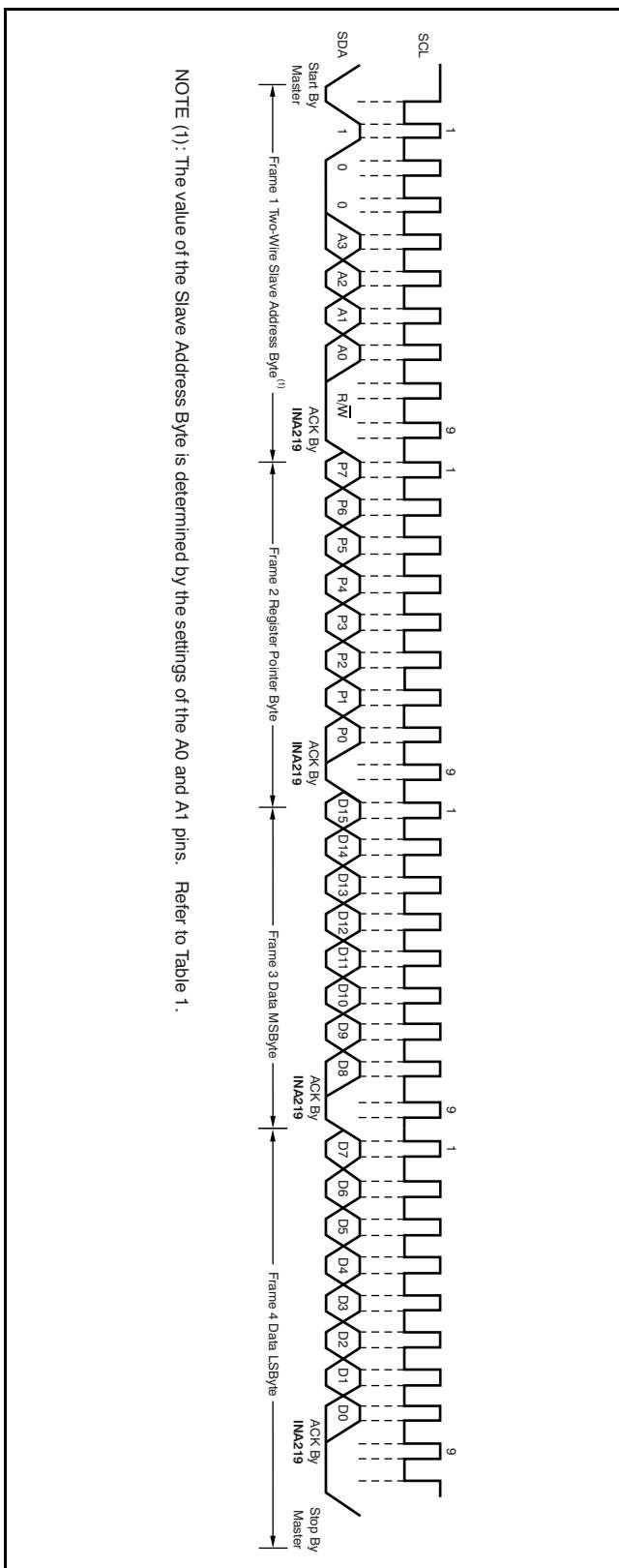


Figure 14. Timing Diagram for Write Word Format

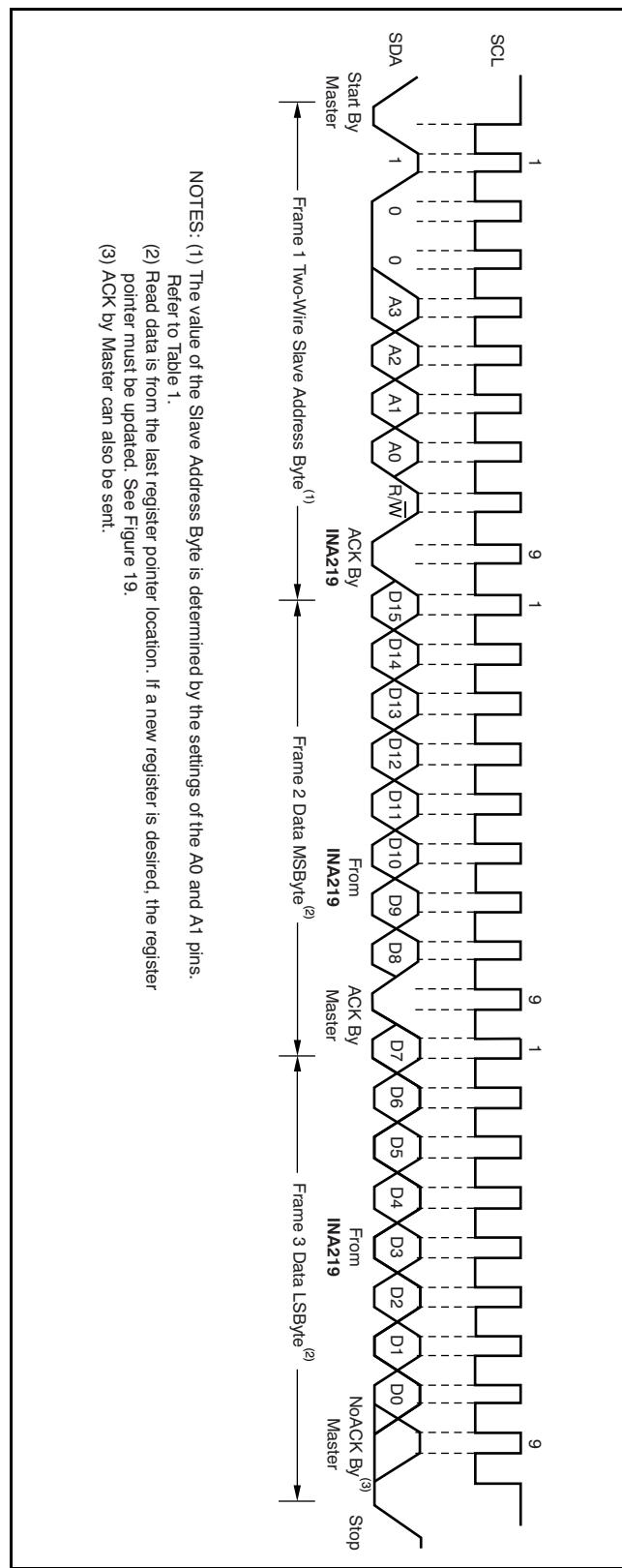


Figure 15. Timing Diagram for Read Word Format

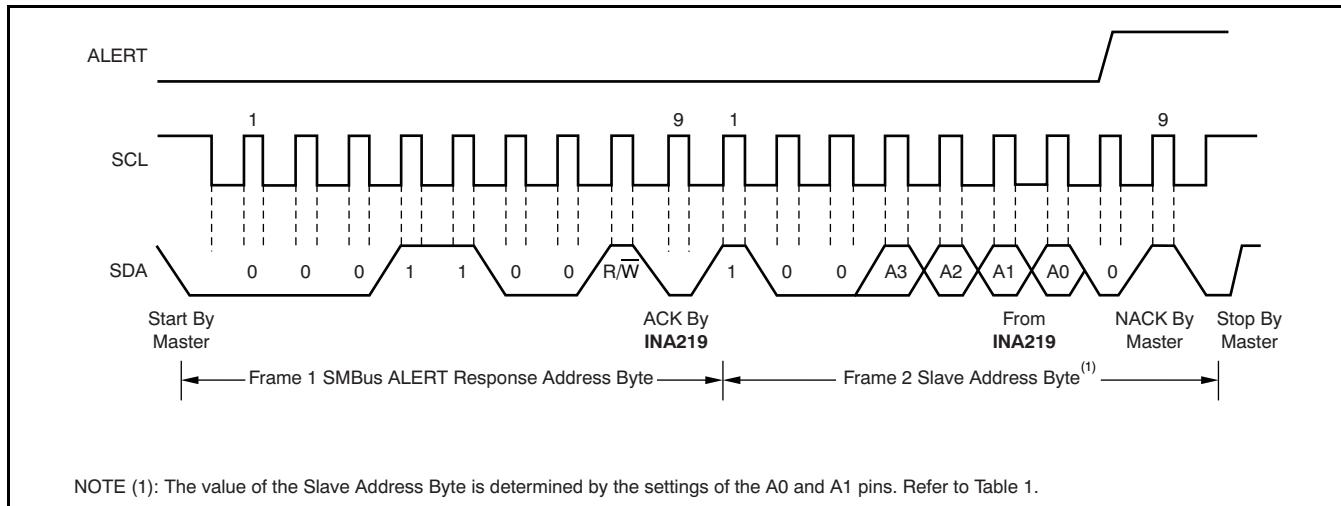


Figure 16. Timing Diagram for SMBus ALERT

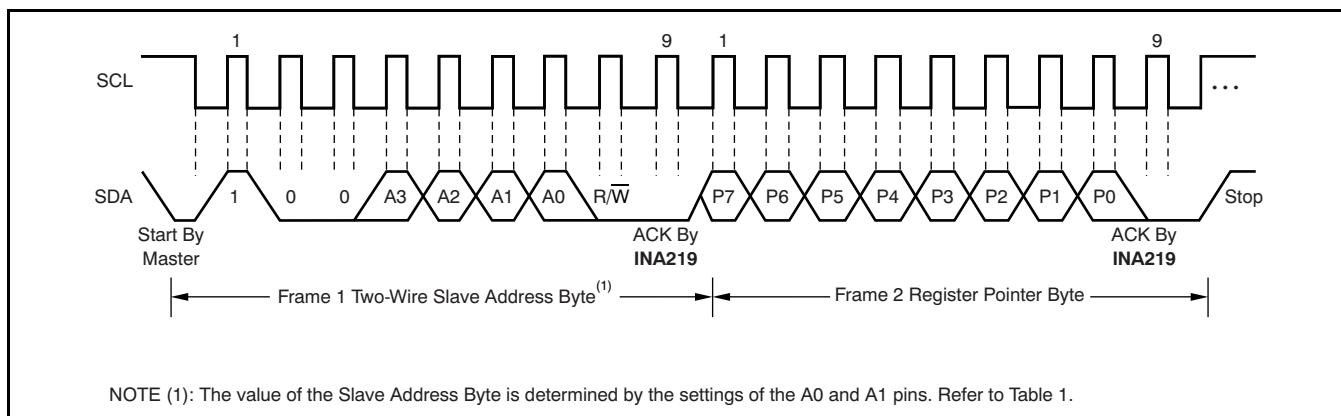
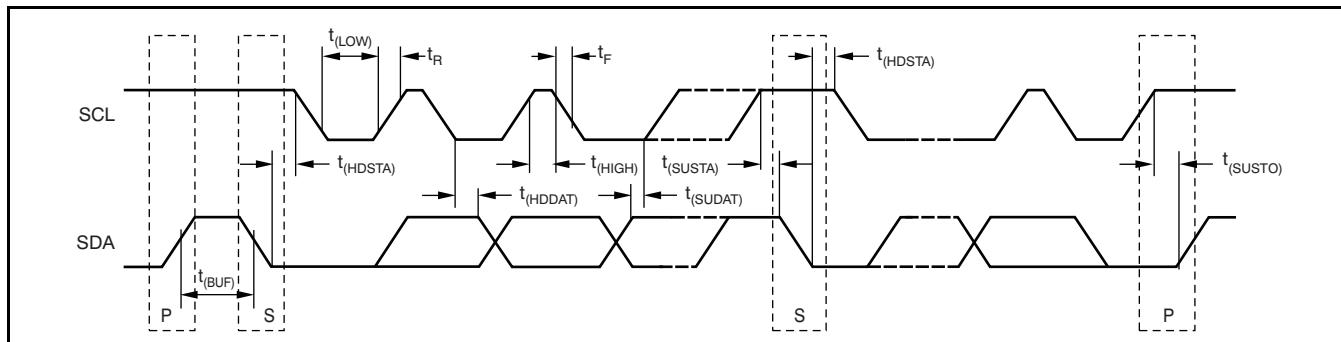


Figure 17. Typical Register Pointer Set

High-Speed I²C Mode

When the bus is idle, both the SDA and SCL lines are pulled high by the pull-up devices. The master generates a start condition followed by a valid serial byte containing High-Speed (HS) master code *00001XXX*. This transmission is made in fast (400kbps) or standard (100kbps) (F/S) mode at no more than 400kbps. The INA219 does not acknowledge the HS master code, but does recognize it and switches its internal filters to support 3.4Mbps operation.

The master then generates a repeated start condition (a repeated start condition has the same timing as the start condition). After this repeated start condition, the protocol is the same as F/S mode, except that transmission speeds up to 3.4Mbps are allowed. Instead of using a stop condition, repeated start conditions should be used to secure the bus in HS-mode. A stop condition ends the HS-mode and switches all the internal filters of the INA219 to support the F/S mode.



Bus Timing Diagram Definitions

PARAMETER	FAST MODE		HIGH-SPEED MODE		UNITS	
	MIN	MAX	MIN	MAX		
SCL Operating Frequency	$f_{(SCL)}$	0.001	0.4	0.001	3.4	MHz
Bus Free Time Between STOP and START Condition	$t_{(BUF)}$	600		160		ns
Hold time after repeated START condition. After this period, the first clock is generated.	$t_{(HDSTA)}$	100		100		ns
Repeated START Condition Setup Time	$t_{(SUSTA)}$	100		100		ns
STOP Condition Setup Time	$t_{(SUSTO)}$	100		100		ns
Data Hold Time	$t_{(HDDAT)}$	0		0		ns
Data Setup Time	$t_{(SUDAT)}$	100		10		ns
SCL Clock LOW Period	$t_{(LOW)}$	1300		160		ns
SCL Clock HIGH Period	$t_{(HIGH)}$	600		60		ns
Clock/Data Fall Time	t_F		300		160	ns
Clock/Data Rise Time	t_R		300		160	ns
Clock/Data Rise Time for SCLK \leq 100kHz	t_R		1000			ns

Power-Up Conditions

Power-up conditions apply to a software reset via the RST bit (bit 15) in the Configuration Register, or the I²C bus General Call Reset.

BASIC ADC FUNCTIONS

The two analog inputs to the INA219, V_{IN+} and V_{IN-}, connect to a shunt resistor in the bus of interest. The INA219 is typically powered by a separate supply from +3V to +5.5V. The bus being sensed can vary from 0V to 26V. There are no special considerations for power-supply sequencing (for example, a bus voltage can be present with the supply voltage off, and vice-versa). The INA219 senses the small drop across the shunt for shunt voltage, and senses the voltage with respect to ground from V_{IN-} for the bus voltage. [Figure 19](#) illustrates this operation.

When the INA219 is in the normal operating mode (that is, MODE bits of the Configuration Register are set to '111'), it continuously converts the shunt voltage up to the number set in the shunt voltage averaging function (Configuration Register, SADC bits). The device then converts the bus voltage up to the number set in the bus voltage averaging (Configuration Register, BADC bits). The Mode control in the Configuration Register also permits selecting modes to convert only voltage or current, either continuously or in response to an event (triggered).

All current and power calculations are performed in the background and do not contribute to conversion time; conversion times shown in the [Electrical Characteristics](#) table can be used to determine the actual conversion time.

Power-Down mode reduces the quiescent current and turns off current into the INA219 inputs, avoiding any supply drain. Full recovery from Power-Down requires 40μs. ADC Off mode (set by the Configuration Register, MODE bits) stops all conversions.

In triggered mode, the external Convert line becomes active. Convert commands are initiated by taking the Convert line low for a minimum of 4μs. The Convert line may be connected high when unused. Any re-trigger of the Convert line during a conversion is ignored, and the Convert line state is disregarded until the conversion ends. There are several available triggered modes; however, all conversions are performed repeatedly up to the number set in the Averaging function (Configuration Register, BADC and SADC bits).

If the Convert line is held low, writing any of the triggered convert modes into the Configuration Register (even if the desired mode is already programmed into the register) triggers a single-shot conversion.

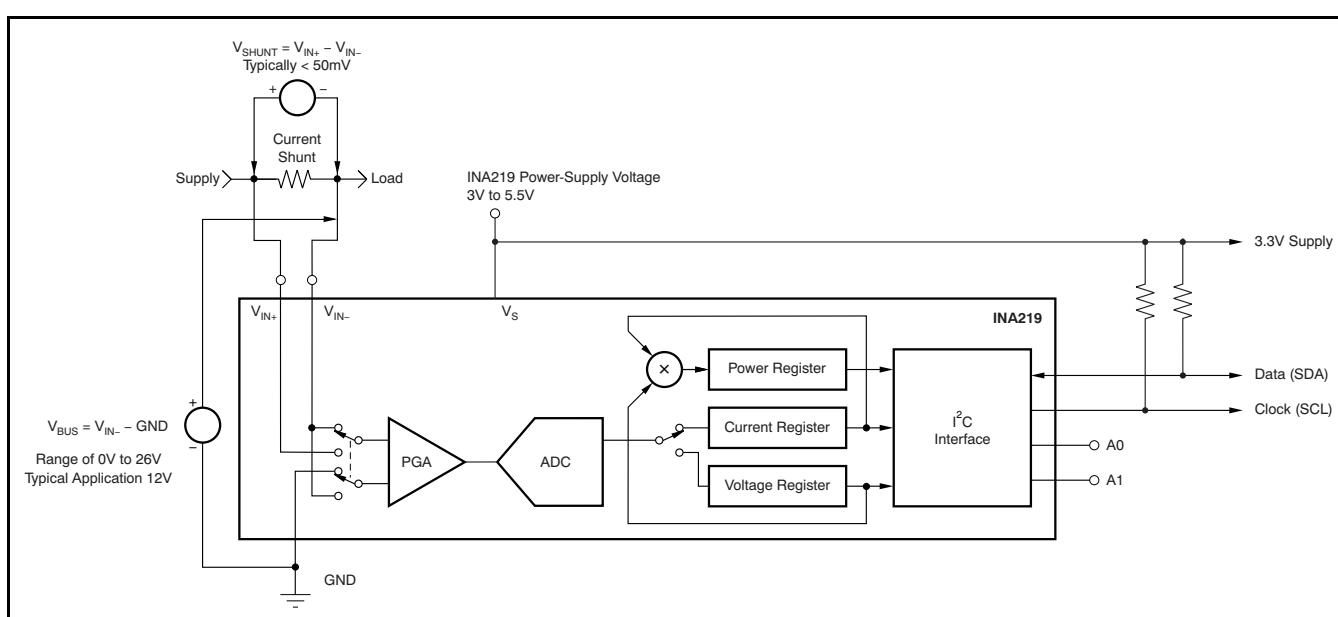


Figure 19. INA219 Configured for Shunt and Bus Voltage Measurement

Although the INA219 can be read at any time, and the data from the last conversion remain available, the Conversion Ready bit (Status Register, CNVR bit) is provided to help co-ordinate one-shot or triggered conversions. The Conversion Ready bit is set after all conversions, averaging, and multiplication operations are complete.

The Conversion Ready bit clears under these conditions:

1. Writing to the Configuration Register, except when configuring the MODE bits for Power Down or ADC off (Disable) modes;
2. Reading the Status Register; or
3. Triggering a single-shot conversion with the Convert pin.

Power Measurement

Current and bus voltage are converted at different points in time, depending on the resolution and averaging mode settings. For instance, when configured for 12-bit and 128 sample averaging, up to 68ms in time between sampling these two values is possible. Again, these calculations are performed in the background and do not add to the overall conversion time.

PGA Function

If larger full-scale shunt voltages are desired, the INA219 provides a PGA function that increases the full-scale range up to 2, 4, or 8 times (320mV). Additionally, the bus voltage measurement has two full-scale ranges: 16V or 32V.

Compatibility with TI Hot Swap Controllers

The INA219 is designed for compatibility with hot swap controllers such the TI [TPS2490](#). The TPS2490 uses a high-side shunt with a limit at 50mV; the INA219 full-scale range of 40mV enables the use of the same shunt for current sensing below this limit. When sensing is required at (or through) the 50mV sense point of the TPS2490, the PGA of the INA219 can be set to ± 2 to provide an 80mV full-scale range.

Filtering and Input Considerations

Measuring current is often noisy, and such noise can be difficult to define. The INA219 offers several options for filtering by choosing resolution and averaging in the Configuration Register. These filtering options can be set independently for either voltage or current measurement.

The internal ADC is based on a delta-sigma ($\Delta\Sigma$) front-end with a 500kHz ($\pm 30\%$) typical sampling rate. This architecture has good inherent noise rejection; however, transients that occur at or very close to the sampling rate harmonics can cause problems. Because these signals are at 1MHz and higher, they can be dealt with by incorporating filtering at the input of the INA219. The high frequency enables the use of low-value series resistors on the filter for negligible effects on measurement accuracy. In general, filtering the INA219 input is only necessary if there are transients at exact harmonics of the 500kHz ($\pm 30\%$) sampling rate ($>1\text{MHz}$). Filter using the lowest possible series resistance and ceramic capacitor. Recommended values are $0.1\mu\text{F}$ to $1.0\mu\text{F}$. [Figure 20](#) shows the INA219 with an additional filter added at the input.

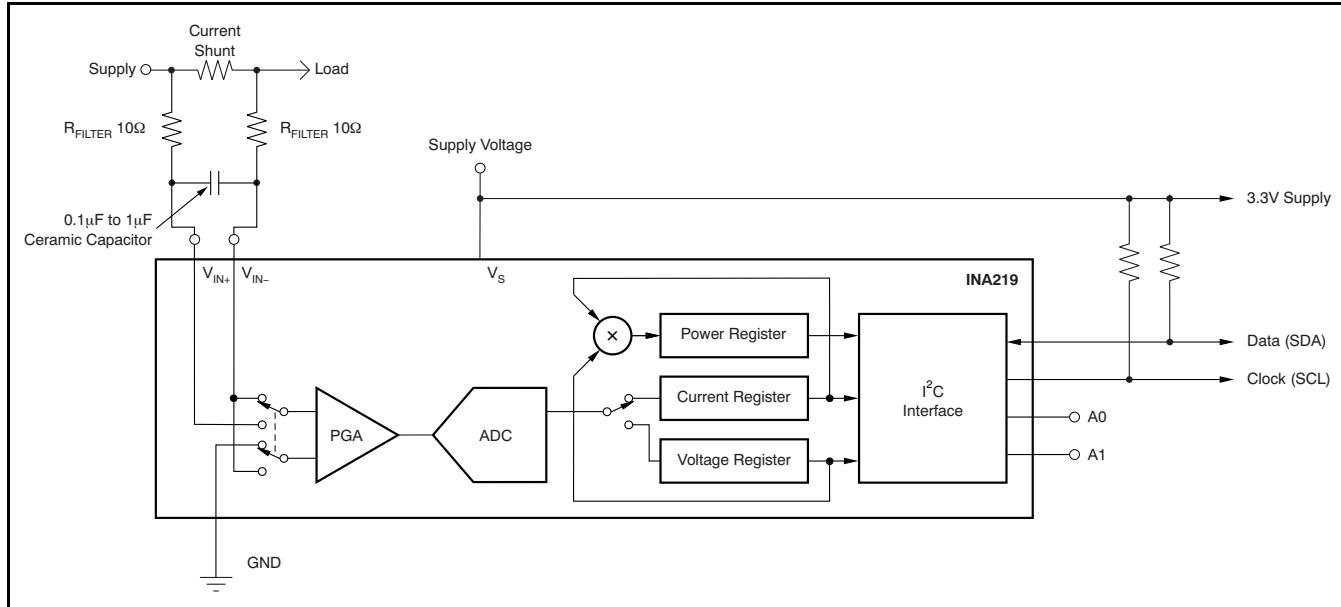


Figure 20. INA219 with Input Filtering

Overload conditions are another consideration for the INA219 inputs. The INA219 inputs are specified to tolerate 26V across the inputs. A large differential scenario might be a short to ground on the load side of the shunt. This type of event can result in full power-supply voltage across the shunt (as long the power supply or energy storage capacitors support it). It must be remembered that removing a short to ground can result in inductive kickbacks that could exceed the 26V differential and common-mode rating of the INA219. Inductive kickback voltages are best dealt with by zener-type transient-absorbing devices (commonly called *transzorbs*) combined with sufficient energy storage capacitance.

In applications that do not have large energy storage electrolytics on one or both sides of the shunt, an input overstress condition may result from an excessive dV/dt of the voltage applied to the input. A hard physical short is the most likely cause of this event, particularly in applications with no large electrolytics present. This problem occurs because an excessive dV/dt can activate the ESD protection in the INA219 in systems where large currents are available. Testing has demonstrated that the addition of 10Ω resistors in series with each input of the INA219 sufficiently protects the inputs against dV/dt failure up to the 26V rating of the INA219. These resistors have no significant effect on accuracy.

Simple Current Shunt Monitor Usage (No Programming Necessary)

The INA219 can be used without any programming if it is only necessary to read a shunt voltage drop and bus voltage with the default 12-bit resolution, 320mV shunt full-scale range (PGA=8), 32V bus full-scale range, and continuous conversion of shunt and bus voltage.

Without programming, current is measured by reading the shunt voltage. The Current Register and Power Register are only available if the Calibration Register contains a programmed value.

Programming the INA219

The default power-up states of the registers are shown in the [INA219 register descriptions](#) section of this data sheet. These registers are volatile, and if programmed to other than default values, must be re-programmed at every device power-up. Detailed information on programming the Calibration Register specifically is given in the section, [Programming the INA219 Power Measurement Engine](#).

PROGRAMMING THE INA219 POWER MEASUREMENT ENGINE

Calibration Register and Scaling

The Calibration Register makes it possible to set the scaling of the Current and Power Registers to whatever values are most useful for a given application. One strategy may be to set the Calibration Register such that the largest possible number is generated in the Current Register or Power Register at the expected full-scale point; this approach yields the highest resolution. The Calibration Register can also be selected to provide values in the Current and Power Registers that either provide direct decimal equivalents of the values being measured, or yield a round LSB number. After these choices have been made, the Calibration Register also offers possibilities for end user system-level calibration, where the value is adjusted slightly to cancel total system error.

Below are two examples for configuring the INA219 calibration. Both examples are written so the information directly relates to the calibration setup found in the INA219EVM software.

Calibration Example 1: Calibrating the INA219 with no possibility for overflow. (Note that the numbers used in this example are the same used with the INA219EVM software as shown in [Figure 21](#).)

- Establish the following parameters:

$$V_{BUS_MAX} = 32$$

$$V_{SHUNT_MAX} = 0.32$$

$$R_{SHUNT} = 0.5$$

- Using [Equation 1](#), determine the maximum possible current .

$$\text{MaxPossible_I} = \frac{V_{SHUNT_MAX}}{R_{SHUNT}}$$

$$\text{MaxPossible_I} = 0.64$$

(1)

- Choose the desired maximum current value. This value is selected based on system expectations.

$$\text{Max_Expected_I} = 0.6$$

- Calculate the possible range of current LSBs. To calculate this range, first compute a range of LSBs that is appropriate for the design. Next, select an LSB within this range. Note that the results will have the most resolution when the minimum LSB is selected. Typically, an LSB is selected to be the nearest round number to the minimum LSB value.

$$\text{Minimum_LSB} = \frac{\text{Max_Expected_I}}{32767}$$

$$\text{Minimum_LSB} = 18.311 \times 10^{-6}$$

(2)

$$\text{Maximum_LSB} = \frac{\text{Max_Expected_I}}{4096}$$

$$\text{Maximum_LSB} = 146.520 \times 10^{-6}$$

(3)

Choose an LSB in the range: $\text{Minimum_LSB} < \text{Selected_LSB} < \text{Maximum_LSB}$

$$\text{Current_LSB} = 20 \times 10^{-6}$$

Note:

This value was selected to be a round number near the Minimum_LSB. This selection allows for good resolution with a rounded LSB.

- Compute the Calibration Register value using [Equation 4](#):

$$\text{Cal} = \text{trunc} \left(\frac{0.04096}{\text{Current_LSB} \times R_{SHUNT}} \right)$$

$$\text{Cal} = 4096$$

(4)

6. Calculate the Power LSB, using [Equation 5](#). [Equation 5](#) shows a general formula; because the bus voltage measurement LSB is always 4mV, the power formula reduces to the calculated result.

Power_LSB = 20 Current_LSB

$$\text{Power_LSB} = 400 \times 10^{-6} \quad (5)$$

7. Compute the maximum current and shunt voltage values (before overflow), as shown by [Equation 6](#) and [Equation 7](#). Note that both [Equation 6](#) and [Equation 7](#) involve an *If - then* condition:

Max_Current = Current_LSB × 32767

$$\text{Max_Current} = 0.65534 \quad (6)$$

If Max_Current ≥ Max_Possible_I then

$$\text{Max_Current_Before_Overflow} = \text{MaxPossible_I}$$

Else

$$\text{Max_Current_Before_Overflow} = \text{Max_Current}$$

End If

(Note that Max_Current is greater than MaxPossible_I in this example.)

Max_Current_Before_Overflow = 0.64 (Note: This result is displayed by software as seen in [Figure 21](#).)

Max_ShuntVoltage = Max_Current_Before_Overflow × R_SHUNT

$$\text{Max_ShuntVoltage} = 0.32 \quad (7)$$

If Max_ShuntVoltage ≥ V_SHUNT_MAX

$$\text{Max_ShuntVoltage_Before_Overflow} = V_{\text{SHUNT_MAX}}$$

Else

$$\text{Max_ShuntVoltage_Before_Overflow} = \text{Max_ShuntVoltage}$$

End If

(Note that Max_ShuntVoltage is greater than V_SHUNT_MAX in this example.)

Max_ShuntVoltage_Before_Overflow = 0.32 (Note: This result is displayed by software as seen in [Figure 21](#).)

8. Compute the maximum power with [Equation 8](#).

MaximumPower = Max_Current_Before_Overflow × V_BUS_MAX

$$\text{MaximumPower} = 20.48 \quad (8)$$

9. (Optional second Calibration step.) Compute corrected full-scale calibration value based on measured current.

INA219_Current = 0.63484

MeaShuntCurrent = 0.55

$$\text{Corrected_Full_Scale_Cal} = \text{trunc} \left(\frac{\text{Cal} \times \text{MeaShuntCurrent}}{\text{INA219_Current}} \right)$$

Corrected_Full_Scale_Cal = 3548

(9)

[Figure 21](#) illustrates how to perform the same procedure discussed in this example using the automated INA219EVM software. Note that the same numbers used in the nine-step example are used in the software example in [Figure 21](#). Also note that [Figure 21](#) illustrates which results correspond to which step (for example, the information entered in Step 1 is enclosed in a box in [Figure 21](#) and labeled).

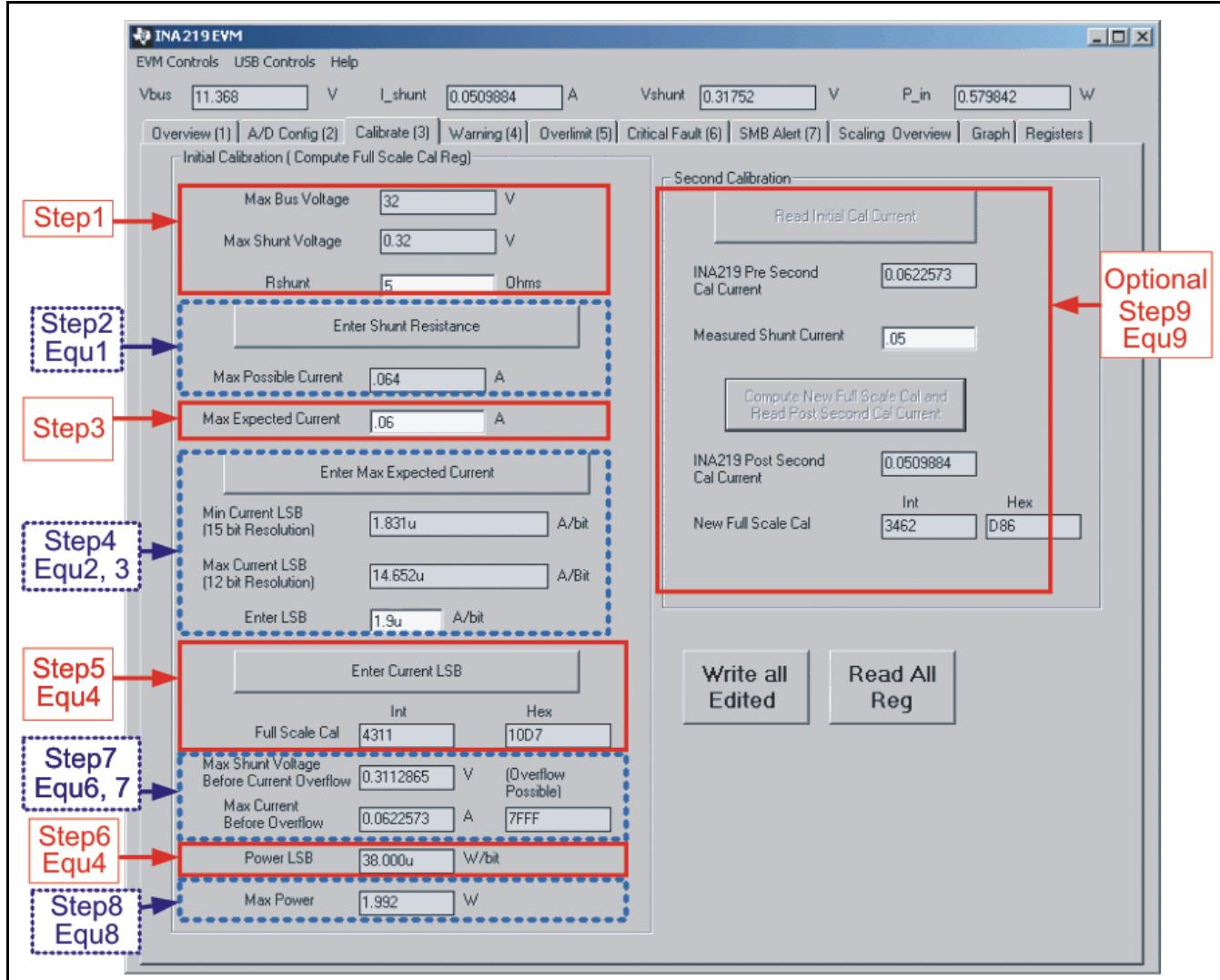


Figure 21. INA219 Calibration Software Automatically Computes Calibration Steps 1-9

Calibration Example 2 (Overflow Possible)

This design example uses the nine-step procedure for calibrating the INA219 where overflow is possible. [Figure 22](#) illustrates how the same procedure is performed using the automated INA219EVM software. Note that the same numbers used in the nine-step example are used in the software example in [Figure 22](#). Also note that [Figure 22](#) illustrates which results correspond to which step (for example, the information entered in Step 1 is circled in [Figure 22](#) and labeled).

- Establish the following parameters:

$$V_{BUS_MAX} = 32$$

$$V_{SHUNT_MAX} = 0.32$$

$$R_{SHUNT} = 5$$

- Determine the maximum possible current using [Equation 10](#):

$$\text{MaxPossible_I} = \frac{V_{SHUNT_MAX}}{R_{SHUNT}}$$

$$\text{MaxPossible_I} = 0.064 \quad (10)$$

- Choose the desired maximum current value: Max_Expected_I , $\leq \text{MaxPossible_I}$. This value is selected based on system expectations.

$$\text{Max_Expected_I} = 0.06$$

- Calculate the possible range of current LSBs. This calculation is done by first computing a range of LSB's that is appropriate for the design. Next, select an LSB within this range. Note that the results will have the most resolution when the minimum LSB is selected. Typically, an LSB is selected to be the nearest round number to the minimum LSB.

$$\text{Minimum_LSB} = \frac{\text{Max_Expected_I}}{32767}$$

$$\text{Minimum_LSB} = 1.831 \times 10^{-6} \quad (11)$$

$$\text{Maximum_LSB} = \frac{\text{Max_Expected_I}}{4096}$$

$$\text{Maximum_LSB} = 14.652 \times 10^{-6} \quad (12)$$

Choose an LSB in the range: $\text{Minimum_LSB} < \text{Selected_LSB} < \text{Maximum_LSB}$

$$\text{Current_LSB} = 1.9 \times 10^{-6}$$

Note:

This value was selected to be a round number near the Minimum_LSB. This section allows for good resolution with a rounded LSB.

- Compute the calibration register using [Equation 13](#):

$$\text{Cal} = \text{trunc} \left(\frac{0.04096}{\text{Current_LSB} \times R_{SHUNT}} \right) \quad \text{Cal} = 4311 \quad (13)$$

- Calculate the Power LSB using [Equation 14](#). [Equation 14](#) shows a general formula; because the bus voltage measurement LSB is always 4mV, the power formula reduces to calculate the result.

$$\text{Power_LSB} = 20 \text{ Current_LSB}$$

$$\text{Power_LSB} = 38 \times 10^{-6} \quad (14)$$

7. Compute the maximum current and shunt voltage values (before overflow), as shown by [Equation 15](#) and [Equation 16](#). Note that both [Equation 15](#) and [Equation 16](#) involve an *If - then* condition.

Max_Current = Current_LSB × 32767

Max_Current = 0.06226 (15)

```
If Max_Current ≥ Max_Possible_I then
    Max_Current_Before_Overflow = MaxPossible_I
Else
    Max_Current_Before_Overflow = Max_Current
End If
```

(Note that Max_Current is less than MaxPossible_I in this example.)

Max_Current_Before_Overflow = 0.06226 (Note: This result is displayed by software as seen in [Figure 22](#).)

Max_ShuntVoltage = Max_Current_Before_Overflow × R_{SHUNT}

Max_ShuntVoltage = 0.3113 (16)

```
If Max_ShuntVoltage ≥ VSHUNT_MAX
    Max_ShuntVoltage_Before_Overflow = VSHUNT_MAX
Else
    Max_ShuntVoltage_Before_Overflow = Max_ShuntVoltage
End If
```

(Note that Max_ShuntVoltage is less than V_{SHUNT_MAX} in this example.)

Max_ShuntVoltage_Before_Overflow = 0.3113 (Note: This result is displayed by software as seen in [Figure 22](#).)

8. Compute the maximum power with equation 8.

MaximumPower = Max_Current_Before_Overflow × V_{BUS_MAX}

MaximumPower = 1.992 (17)

9. (Optional second calibration step.) Compute the corrected full-scale calibration value based on measured current.

INA219_Current = 0.06226

MeaShuntCurrent = 0.05

Corrected_Full_Scale_Cal = trunc $\left\lfloor \frac{\text{Cal} \times \text{MeasShuntCurrent}}{\text{INA219_Current}} \right\rfloor$

Corrected_Full_Scale_Cal = 3462 (18)

[Figure 22](#) illustrates how to perform the same procedure discussed in this example using the automated INA219EVM software. Note that the same numbers used in the nine-step example are used in the software example in [Figure 22](#). Also note that [Figure 22](#) illustrates which results correspond to which step (for example, the information entered in Step 1 is enclosed in a box in [Figure 22](#) and labeled).

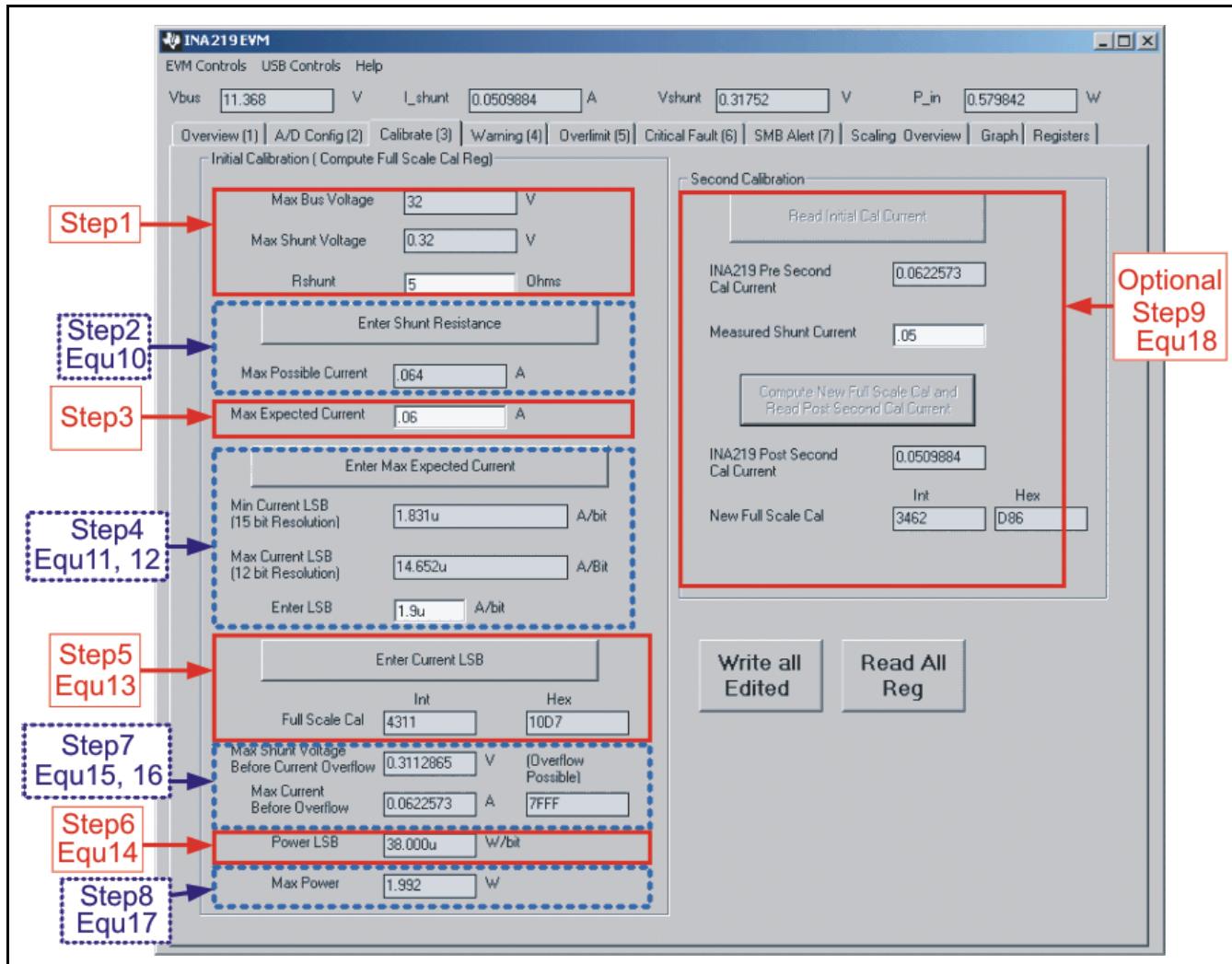


Figure 22. Calibration Software Automatically Computes Calibration Steps 1-9

REGISTER INFORMATION

The INA219 uses a bank of registers for holding configuration settings, measurement results, maximum/minimum limits, and status information. [Table 2](#) summarizes the INA219 registers; [Figure 12](#) illustrates registers.

Register contents are updated 4 μ s after completion of the write command. Therefore, a 4 μ s delay is required between completion of a write to a given register and a subsequent read of that register (without changing the pointer) when using SCL frequencies in excess of 1MHz.

Table 2. Summary of Register Set

POINTER ADDRESS	REGISTER NAME	FUNCTION	POWER-ON RESET		TYPE ⁽¹⁾
			BINARY	HEX	
00	Configuration Register	All-register reset, settings for bus voltage range, PGA Gain, ADC resolution/averaging.	0011001 10011111	399F	R/W
01	Shunt Voltage	Shunt voltage measurement data.	Shunt voltage	—	R
02	Bus Voltage	Bus voltage measurement data.	Bus voltage	—	R
03	Power ⁽²⁾	Power measurement data.	00000000 00000000	0000	R
04	Current ⁽²⁾	Contains the value of the current flowing through the shunt resistor.	00000000 00000000	0000	R
05	Calibration	Sets full-scale range and LSB of current and power measurements. Overall system calibration.	00000000 00000000	0000	R/W

(1) Type: R = Read-Only, R/W = Read/Write.

(2) The Power Register and Current Register default to '0' because the Calibration Register defaults to '0', yielding a zero current value until the Calibration Register is programmed.

REGISTER DETAILS

All INA219 registers 16-bit registers are actually two 8-bit bytes via the I²C interface.

Configuration Register 00h (Read/Write)

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	RST	—	BRNG	PG1	PG0	BADC4	BADC3	BADC2	BADC1	SADC4	SADC3	SADC2	SADC1	MODE3	MODE2	MODE1
POR VALUE	0	0	1	1	0	0	1	1	0	0	1	1	1	1	1	1

Bit Descriptions

RST: **Reset Bit**

Bit 15 Setting this bit to '1' generates a system reset that is the same as power-on reset. Resets all registers to default values; this bit self-clears.

BRNG: **Bus Voltage Range**

Bit 13
0 = 16V FSR
1 = 32V FSR (default value)

PG: **PGA (Shunt Voltage Only)**

Bits 11, 12 Sets PGA gain and range. Note that the PGA defaults to $\div 8$ (320mV range). [Table 3](#) shows the gain and range for the various product gain settings.

Table 3. PG Bit Settings⁽¹⁾

PG1	PG0	GAIN	RANGE
0	0	1	$\pm 40\text{mV}$
0	1	$\div 2$	$\pm 80\text{mV}$
1	0	$\div 4$	$\pm 160\text{mV}$
1	1	$\div 8$	$\pm 320\text{mV}$

(1) Shaded values are default.

BADC: **BADC Bus ADC Resolution/Averaging**

Bits 7–10 These bits adjust the Bus ADC resolution (9-, 10-, 11-, or 12-bit) or set the number of samples used when averaging results for the Bus Voltage Register (02h).

SADC: **SADC Shunt ADC Resolution/Averaging**

Bits 3–6

These bits adjust the Shunt ADC resolution (9-, 10-, 11-, or 12-bit) or set the number of samples used when averaging results for the Shunt Voltage Register (01h).

BADC (Bus) and SADC (Shunt) ADC resolution/averaging and conversion time settings are shown in [Table 4](#).

Table 4. ADC Settings⁽¹⁾

ADC4	ADC3	ADC2	ADC1	MODE/SAMPLES	CONVERSION TIME
0	X ⁽²⁾	0	0	9-bit	84µs
0	X ⁽²⁾	0	1	10-bit	148µs
0	X ⁽²⁾	1	0	11-bit	276µs
0	X ⁽²⁾	1	1	12-bit	532µs
1	0	0	0	12-bit	532µs
1	0	0	1	2	1.06ms
1	0	1	0	4	2.13ms
1	0	1	1	8	4.26ms
1	1	0	0	16	8.51ms
1	1	0	1	32	17.02ms
1	1	1	0	64	34.05ms
1	1	1	1	128	68.10ms

(1) Shaded values are default.

(2) X = Don't care.

MODE: **Operating Mode**

Bits 0–2

Selects continuous, triggered, or power-down mode of operation. These bits default to continuous shunt and bus measurement mode. The mode settings are shown in [Table 5](#).

Table 5. Mode Settings⁽¹⁾

MODE3	MODE2	MODE1	MODE
0	0	0	Power-Down
0	0	1	Shunt Voltage, Triggered
0	1	0	Bus Voltage, Triggered
0	1	1	Shunt and Bus, Triggered
1	0	0	ADC Off (disabled)
1	0	1	Shunt Voltage, Continuous
1	1	0	Bus Voltage, Continuous
1	1	1	Shunt and Bus, Continuous

(1) Shaded values are default.

DATA OUTPUT REGISTERS

Shunt Voltage Register 01h (Read-Only)

The Shunt Voltage Register stores the current shunt voltage reading, V_{SHUNT} . Shunt Voltage Register bits are shifted according to the PGA setting selected in the Configuration Register (00h). When multiple sign bits are present, they will all be the same value. Negative numbers are represented in two's complement format. Generate the two's complement of a negative number by complementing the absolute value binary number and adding 1. Extend the sign, denoting a negative number by setting the MSB = '1'. Extend the sign to any additional sign bits to form the 16-bit word.

Example: For a value of $V_{SHUNT} = -320\text{mV}$:

1. Take the absolute value (include accuracy to 0.01mV) ==> 320.00
2. Translate this number to a whole decimal number ==> 32000
3. Convert it to binary ==> 111 1101 0000 0000
4. Complement the binary result : 000 0010 1111 1111
5. Add 1 to the Complement to create the Two's Complement formatted result ==> 000 0011 0000 0000
6. Extend the sign and create the 16-bit word: 1000 0011 0000 0000 = 8300h (Remember to extend the sign to all sign-bits, as necessary based on the PGA setting.)

At PGA = ± 8 , full-scale range = $\pm 320\text{mV}$ (decimal = 32000, positive value hex = 7D00, negative value hex = 8300), and LSB = $10\mu\text{V}$.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	SIGN	SD14_8	SD13_8	SD12_8	SD11_8	SD10_8	SD9_8	SD8_8	SD7_8	SD6_8	SD5_8	SD4_8	SD3_8	SD2_8	SD1_8	SD0_8
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

At PGA = ± 4 , full-scale range = $\pm 160\text{mV}$ (decimal = 16000, positive value hex = 3E80, negative value hex = C180), and LSB = $10\mu\text{V}$.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	SIGN	SIGN	SD13_4	SD12_4	SD11_4	SD10_4	SD9_4	SD8_4	SD7_4	SD6_4	SD5_4	SD4_4	SD3_4	SD2_4	SD1_4	SD0_4
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

At PGA = ± 2 , full-scale range = $\pm 80\text{mV}$ (decimal = 8000, positive value hex = 1F40, negative value hex = E0C0), and LSB = $10\mu\text{V}$.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	SIGN	SIGN	SIGN	SD12_2	SD11_2	SD10_2	SD9_2	SD8_2	SD7_2	SD6_2	SD5_2	SD4_2	SD3_2	SD2_2	SD1_2	SD0_2
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

At PGA = ± 1 , full-scale range = $\pm 40\text{mV}$ (decimal = 4000, positive value hex = 0FA0, negative value hex = F060), and LSB = $10\mu\text{V}$.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	SIGN	SIGN	SIGN	SIGN	SD11_1	SD10_1	SD9_1	SD8_1	SD7_1	SD6_1	SD5_1	SD4_1	SD3_1	SD2_1	SD1_1	SD0_1
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6. Shunt Voltage Register Format⁽¹⁾

V_{SHUNT} Reading (mV)	Decimal Value	PGA = $\div 8$ (D15.....D0)	PGA = $\div 4$ (D15.....D0)	PGA = $\div 2$ (D15.....D0)	PGA = $\div 1$ (D15.....D0)
320.02	32002	0111 1101 0000 0000	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
320.01	32001	0111 1101 0000 0000	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
320.00	32000	0111 1101 0000 0000	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
319.99	31999	0111 1100 1111 1111	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
319.98	31998	0111 1100 1111 1110	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
:	:	:	:	:	:
160.02	16002	0011 1110 1000 0010	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
160.01	16001	0011 1110 1000 0001	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
160.00	16000	0011 1110 1000 0000	0011 1110 1000 0000	0001 1111 0100 0000	0000 1111 1010 0000
159.99	15999	0011 1110 0111 1111	0011 1110 0111 1111	0001 1111 0100 0000	0000 1111 1010 0000
159.98	15998	0011 1110 0111 1110	0011 1110 0111 1110	0001 1111 0100 0000	0000 1111 1010 0000
:	:	:	:	:	:
80.02	8002	0001 1111 0100 0010	0001 1111 0100 0010	0001 1111 0100 0000	0000 1111 1010 0000
80.01	8001	0001 1111 0100 0001	0001 1111 0100 0001	0001 1111 0100 0000	0000 1111 1010 0000
80.00	8000	0001 1111 0100 0000	0001 1111 0100 0000	0001 1111 0100 0000	0000 1111 1010 0000
79.99	7999	0001 1111 0011 1111	0001 1111 0011 1111	0001 1111 0011 1111	0000 1111 1010 0000
79.98	7998	0001 1111 0011 1110	0001 1111 0011 1110	0001 1111 0011 1110	0000 1111 1010 0000
:	:	:	:	:	:
40.02	4002	0000 1111 1010 0010	0000 1111 1010 0010	0000 1111 1010 0010	0000 1111 1010 0000
40.01	4001	0000 1111 1010 0001	0000 1111 1010 0001	0000 1111 1010 0001	0000 1111 1010 0000
40.00	4000	0000 1111 1010 0000	0000 1111 1010 0000	0000 1111 1010 0000	0000 1111 1010 0000
39.99	3999	0000 1111 1001 1111	0000 1111 1001 1111	0000 1111 1001 1111	0000 1111 1001 1111
39.98	3998	0000 1111 1001 1110	0000 1111 1001 1110	0000 1111 1001 1110	0000 1111 1001 1110
:	:	:	:	:	:
0.02	2	0000 0000 0000 0010	0000 0000 0000 0010	0000 0000 0000 0010	0000 0000 0000 0010
0.01	1	0000 0000 0000 0001	0000 0000 0000 0001	0000 0000 0000 0001	0000 0000 0000 0001
0	0	0000 0000 0000 0000	0000 0000 0000 0000	0000 0000 0000 0000	0000 0000 0000 0000
-0.01	-1	1111 1111 1111 1111	1111 1111 1111 1111	1111 1111 1111 1111	1111 1111 1111 1111
-0.02	-2	1111 1111 1111 1110	1111 1111 1111 1110	1111 1111 1111 1110	1111 1111 1111 1110
:	:	:	:	:	:
-39.98	-3998	1111 0000 0110 0010	1111 0000 0110 0010	1111 0000 0110 0010	1111 0000 0110 0010
-39.99	-3999	1111 0000 0110 0001	1111 0000 0110 0001	1111 0000 0110 0001	1111 0000 0110 0001
-40.00	-4000	1111 0000 0110 0000	1111 0000 0110 0000	1111 0000 0110 0000	1111 0000 0110 0000
-40.01	-4001	1111 0000 0101 1111	1111 0000 0101 1111	1111 0000 0101 1111	1111 0000 0110 0000
-40.02	-4002	1111 0000 0101 1110	1111 0000 0101 1110	1111 0000 0101 1110	1111 0000 0110 0000
:	:	:	:	:	:
-79.98	-7998	1110 0000 1100 0010	1110 0000 1100 0010	1110 0000 1100 0010	1111 0000 0110 0000
-79.99	-7999	1110 0000 1100 0001	1110 0000 1100 0001	1110 0000 1100 0001	1111 0000 0110 0000
-80.00	-8000	1110 0000 1100 0000	1110 0000 1100 0000	1110 0000 1100 0000	1111 0000 0110 0000
-80.01	-8001	1110 0000 1011 1111	1110 0000 1011 1111	1110 0000 1100 0000	1111 0000 0110 0000
-80.02	-8002	1110 0000 1011 1110	1110 0000 1011 1110	1110 0000 1100 0000	1111 0000 0110 0000
:	:	:	:	:	:
-159.98	-15998	1100 0001 1000 0010	1100 0001 1000 0010	1110 0000 1100 0000	1111 0000 0110 0000
-159.99	-15999	1100 0001 1000 0001	1100 0001 1000 0001	1110 0000 1100 0000	1111 0000 0110 0000
-160.00	-16000	1100 0001 1000 0000	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000
-160.01	-16001	1100 0001 0111 1111	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000
-160.02	-16002	1100 0001 0111 1110	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000
:	:	:	:	:	:
-319.98	-31998	1000 0011 0000 0010	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000
-319.99	-31999	1000 0011 0000 0001	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000
-320.00	-32000	1000 0011 0000 0000	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000
-320.01	-32001	1000 0011 0000 0000	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000
-320.02	-32002	1000 0011 0000 0000	1100 0001 1000 0000	1110 0000 1100 0000	1111 0000 0110 0000

(1) Out-of-range values are shown in grey shading.

Bus Voltage Register 02h (Read-Only)

The Bus Voltage Register stores the most recent bus voltage reading, V_{BUS} .

At full-scale range = 32V (decimal = 8000, hex = 1F40), and LSB = 4mV.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	BD12	BD11	BD10	BD9	BD8	BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0	—	CNVR	OVF
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

At full-scale range = 16V (decimal = 4000, hex = 0FA0), and LSB = 4mV.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	0	BD11	BD10	BD9	BD8	BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0	—	CNVR	OVF
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CNVR: Conversion Ready

Bit 1

Although the data from the last conversion can be read at any time, the INA219 Conversion Ready bit (CNVR) indicates when data from a conversion is available in the data output registers. The CNVR bit is set after all conversions, averaging, and multiplications are complete. CNVR will clear under the following conditions:

- 1.) Writing a new mode into the Operating Mode bits in the Configuration Register (except for Power-Down or Disable)
- 2.) Reading the Power Register

OVF: Math Overflow Flag

Bit 0

The Math Overflow Flag (OVF) is set when the Power or Current calculations are out of range. It indicates that current and power data may be meaningless.

Power Register 03h (Read-Only)

Full-scale range and LSB are set by the Calibration Register. See the [Programming the INA219 Power Measurement Engine](#) section.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The Power Register records power in watts by multiplying the values of the current with the value of the bus voltage according to the equation:

$$\text{Power} = \frac{\text{Current} \times \text{BusVoltage}}{5000}$$

Current Register 04h (Read-Only)

Full-scale range and LSB depend on the value entered in the Calibration Register. See the [Programming the INA219 Power Measurement Engine](#) section. Negative values are stored in two's complement format.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
BIT NAME	CSIGN	CD14	CD13	CD12	CD11	CD10	CD9	CD8	CD7	CD6	CD5	CD4	CD3	CD2	CD1	CD0
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The value of the Current Register is calculated by multiplying the value in the Shunt Voltage Register with the value in the Calibration Register according to the equation:

$$\text{Current} = \frac{\text{ShuntVoltage} \times \text{Calibration Register}}{4096}$$

CALIBRATION REGISTER

Calibration Register 05h (Read/Write)

Current and power calibration are set by bits D15 to D1 of the Calibration Register. Note that bit D0 is not used in the calculation. This register sets the current that corresponds to a full-scale drop across the shunt. Full-scale range and the LSB of the current and power measurement depend on the value entered in this register. See the [Programming the INA219 Power Measurement Engine](#) section. This register is suitable for use in overall system calibration. Note that the '0' POR values are all default.

BIT #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0 ⁽¹⁾
BIT NAME	FS15	FS14	FS13	FS12	FS11	FS10	FS9	FS8	FS7	FS6	FS5	FS4	FS3	FS2	FS1	FS0
POR VALUE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- (1) D0 is a void bit and will always be '0'. It is not possible to write a '1' to D0. CALIBRATION is the value stored in D15:D1.

PACKAGING INFORMATION

Orderable Device	Status ⁽¹⁾	Package Type	Package Drawing	Pins	Package Qty	Eco Plan ⁽²⁾	Lead/Ball Finish	MSL Peak Temp ⁽³⁾
INA219AID	PREVIEW	SOIC	D	8	75	TBD	Call TI	Call TI
INA219AIDCNR	ACTIVE	SOT-23	DCN	8	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR
INA219AIDCNRG4	ACTIVE	SOT-23	DCN	8	3000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR
INA219AIDCNT	ACTIVE	SOT-23	DCN	8	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR
INA219AIDCNG4	ACTIVE	SOT-23	DCN	8	250	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-2-260C-1 YEAR
INA219AIDR	PREVIEW	SOIC	D	8	2500	TBD	Call TI	Call TI

⁽¹⁾ The marketing status values are defined as follows:

ACTIVE: Product device recommended for new designs.

LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.

NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.

PREVIEW: Device has been announced but is not in production. Samples may or may not be available.

OBSOLETE: TI has discontinued the production of the device.

⁽²⁾ Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check <http://www.ti.com/productcontent> for the latest availability information and additional product content details.

TBD: The Pb-Free/Green conversion plan has not been defined.

Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.

Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.

Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

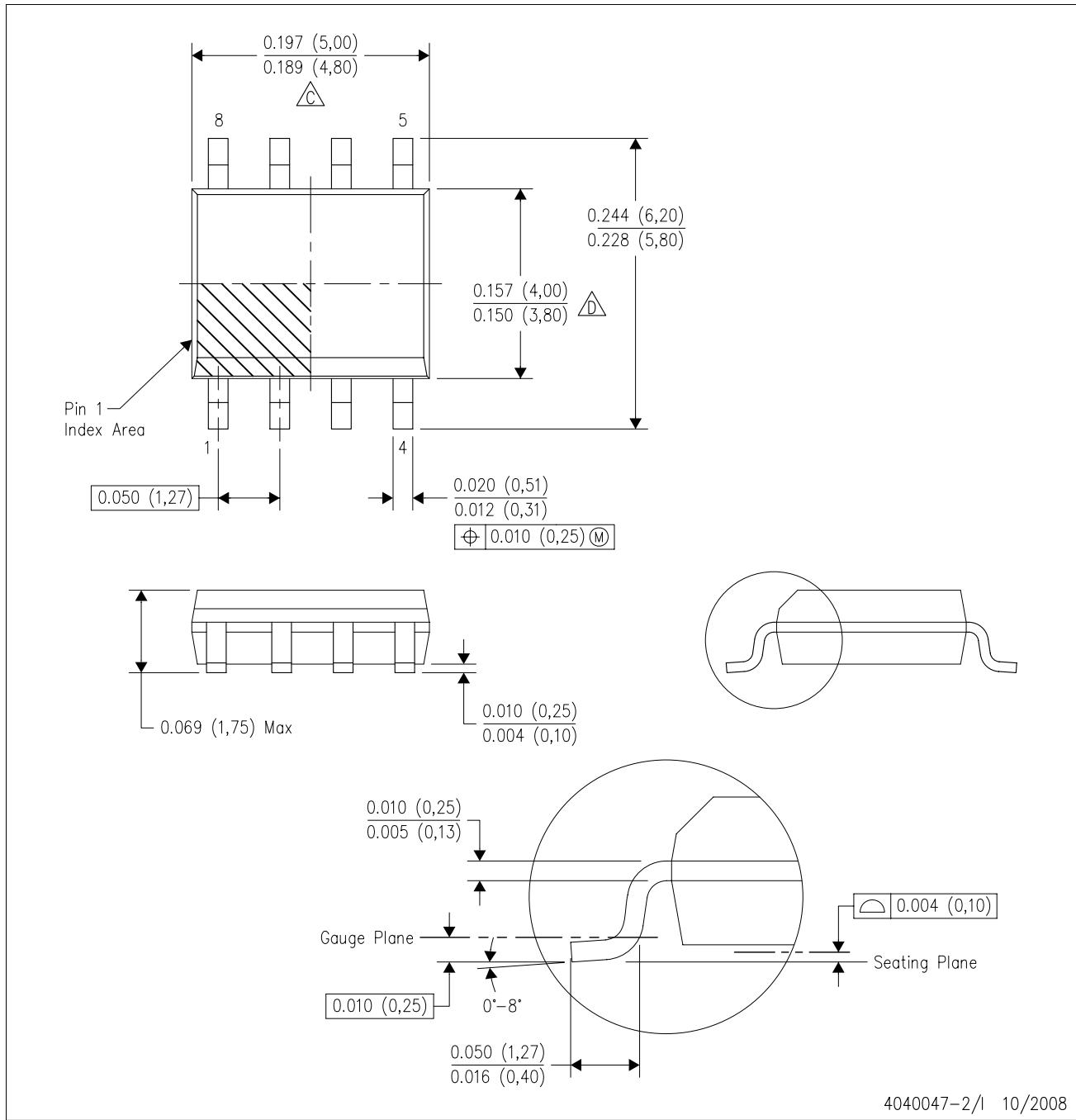
⁽³⁾ MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

D (R-PDSO-G8)

PLASTIC SMALL-OUTLINE PACKAGE



NOTES: A. All linear dimensions are in inches (millimeters).

B. This drawing is subject to change without notice.

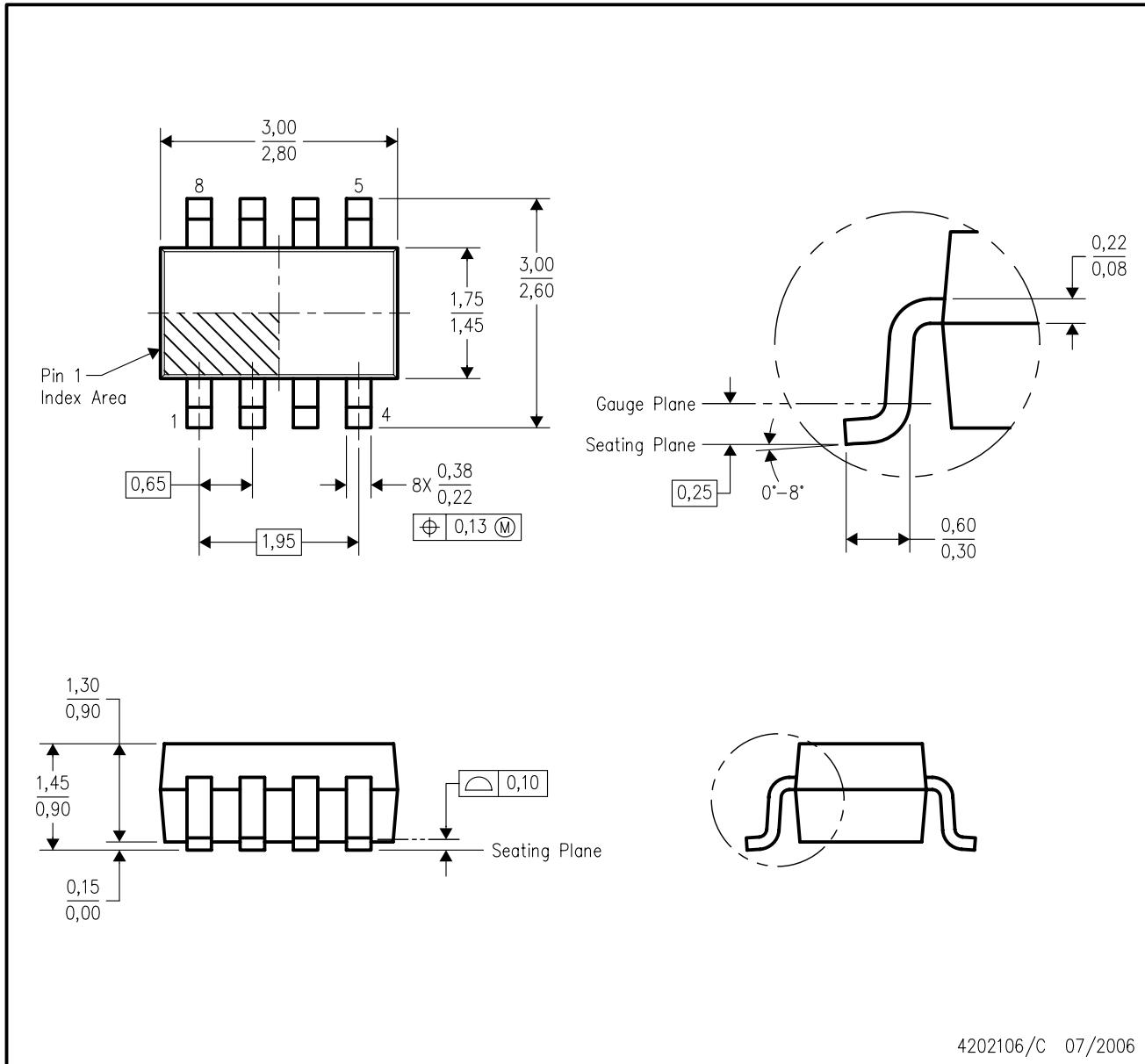
C. Body length does not include mold flash, protrusions, or gate burrs. Mold flash, protrusions, or gate burrs shall not exceed .006 (0,15) per end.

D. Body width does not include interlead flash. Interlead flash shall not exceed .017 (0,43) per side.

E. Reference JEDEC MS-012 variation AA.

DCN (R-PDSO-G8)

PLASTIC SMALL-OUTLINE PACKAGE (DIE DOWN)



- NOTES:
- All linear dimensions are in millimeters.
 - This drawing is subject to change without notice.
 - Package outline exclusive of mold flash, metal burr & dambar protrusion/intrusion.
 - Package outline inclusive of solder plating.
 - A visual index feature must be located within the Pin 1 index area.
 - Falls within JEDEC MO-178 Variation BA.

4202106/C 07/2006

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products

Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
RF/IF and ZigBee® Solutions	www.ti.com/lpr

Applications

Audio	www.ti.com/audio
Automotive	www.ti.com/automotive
Broadband	www.ti.com/broadband
Digital Control	www.ti.com/digitalcontrol
Medical	www.ti.com/medical
Military	www.ti.com/military
Optical Networking	www.ti.com/opticalnetwork
Security	www.ti.com/security
Telephony	www.ti.com/telephony
Video & Imaging	www.ti.com/video
Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2008, Texas Instruments Incorporated