



TUGAS AKHIR - TE 141599

Kontrol Lengan Robot yang Meniru Pergerakan Tangan untuk Inspeksi Objek yang Mengandung Gas Berbahaya

Wahyu Satrio Prayogo
NRP 07111645000047

Dosen Pembimbing
Dr. Muhammad Rivai, ST., MT.
Fajar Budiman, ST., M.Sc.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 141599

Kontrol Lengan Robot yang Meniru Pergerakan Tangan untuk Inspeksi Objek yang Mengandung Gas Berbahaya

Wahyu Satrio Prayogo
NRP 07111645000047

Dosen Pembimbing
Dr. Muhammad Rivai, ST., MT.
Fajar Budiman, ST., M.Sc.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TE 141599

Robotic Arm Control Replicating Hand Movement to Inspect Object Containing Dangerous Gas

Wahyu Satrio Prayogo
NRP 07111645000047

Advisors
Dr. Muhammad Rivai, ST., MT.
Fajar Budiman, ST., M.Sc.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Kontrol Lengan Robot Yang Meniru Pergerakan Tangan Untuk Inspeksi Objek Yang Mengandung Gas Berbahaya**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 26 Juni 2018



Wahyu Satrio Prayogo
NRP 07111645000047

-----Halaman ini sengaja dikosongkan-----

**KONTROL Lengan ROBOT YANG MENIRU
PERGERAKAN TANGAN UNTUK INSPEKSI OBJEK
YANG MENGANDUNG GAS BERBAHAYA**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Teknik Elektronika
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I,

Dosen Pembimbing II,

Dr. Muhammad Rivai, S.T., M.T.
NIP: 196904261994031003

Fajar Budiman, ST., M.Sc.
NIP: 198607072014041001



-----Halaman ini sengaja dikosongkan-----

Kontrol Lengan Robot Yang Meniru Pergerakan Tangan Untuk Inspeksi Objek Yang Mengandung Gas Berbahaya

Nama : Wahyu Satrio Prayogo
Pembimbing I : Dr. Muhammad Rivai, ST., MT.
Pembimbing II : Fajar Budiman, ST., M.Sc.

ABSTRAK

Beberapa tahun terakhir ini banyak sekali teror bom yang ada di Indonesia, hal tersebut menyebabkan ketidaknyamanan orang dalam melakukan aktifitas sehari-hari di tempat umum. Untuk itu pada penelitian ini telah dibuat suatu lengan robot yang dapat dikendalikan berdasarkan pergerakan lengan manusia secara *wireless*. Pergerakan lengan dideteksi oleh sensor IMU MPU6050. Sensor ini dapat mendeteksi perubahan posisi pada sumbu x, y dan z, sehingga lengan robot dapat bergerak sesuai dengan perubahan posisi dari lengan operator. Pergerakan *gripper* dideteksi oleh sensor *flex* yang diletakkan pada jari operator, sehingga dari pergerakan jari dapat mengontrol pergerakan *gripper* yang ada pada lengan robot. Posisi lengan robot dikendalikan menggunakan kontrol Proporsional. Alat ini dilengkapi dengan sensor gas MQ2 yang digunakan sebagai sensor gas berbahaya. Mikrokontroler Arduino Uno digunakan mengakuisisi data sensor gas, menggerakkan motor servo dan sistem pengendali. Sedangkan mikrokontroler Arduino Nano digunakan untuk mengakuisisi data sensor IMU dan Flex. Dari hasil eksperimen menunjukkan bahwa nilai kesalahan posisi pada lengan robot untuk sumbu *pitch* dan *yaw* masing – masing sebesar 4% dan 0,2%. Nilai *Threshold* untuk sensor gas MQ2 adalah 0,33 Volt . Jarak maksimal modul Bluetooth dalam mengirimkan data adalah 5 meter tanpa terhalang apapun. Hasil dari rancang bangun lengan robot ini diharapkan dapat membantu pekerjaan manusia dan juga memperkecil risiko terluka akibat bahan peledak atau gas yang membahayakan manusia.

Kata Kunci : Lengan Robot, MQ2, Mikrokontroler, Sensor IMU

-----Halaman ini sengaja dikosongkan-----

Robotic Arm Control That Replicates Hand Movement to Inspect Object That Contain Dangerous Gas

Name : Wahyu Satrio Prayogo
Advisor 1st : Dr. Muhammad Rivai, ST., MT.
Advisor 2nd : Fajar Budiman, ST., M.Sc.

ABSTRACT

In the last few years there have been so many bombings in Indonesia that caused inconvenience to people in doing their daily activities in public places. Study has been conducted to design a robot arm that can be controlled based on the movement of the human arm wirelessly. The movement of the arm are detected by the IMU MPU6050 sensors. The sensors can detect position changes on the x, y and z axis, so that the robot arm can move according to the change of position of the operator's arm. Gripper movement is detected by the flex sensor placed on the operator's finger, so that the movement of the finger can control the movement of gripper in the robot arm. The position of the robot arm is controlled using Proportional control. The system is equipped with a gas sensor MQ2 which is used as a harmful gas sensor. Arduino Uno microcontroller is used to acquire gas sensor data, drive servo motor and control system. Arduino Nano microcontroller is used to acquire IMU and Flex sensor data. The experimental results show that the position error values on the robot arm for the pitch and yaw axes are 4% and 0.2%, respectively. The Threshold value for the MQ2 gas sensor is 0,33 Volt. The maximum distance of the Bluetooth module in sending data is 5 meters without any hindrance. The robot arm design is expected to help human work and also minimize the risk of injury caused by explosives or gases that harm people.

Keywords : IMU Sensor, Microcontroller, MQ2, Robot Arm

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir dengan judul **Kontrol Lengan Robot Yang Meniru Pergerakan Tangan Untuk Inspeksi Objek Yang Mengandung Gas Berbahaya** dapat terselesaikan dengan baik. Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Strata-1 pada Bidang Studi Teknik Elektronika, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya.

Pada kesempatan ini, penulis ingin berterima kasih kepada pihak-pihak yang membantu pembuatan tugas akhir ini, khususnya kepada :

1. Dr. Muhammad Rivai, S.T., M.T. dan Fajar Budiman, S.T., M.Sc. selaku dosen pembimbing atas bimbingan dan bantuan dalam merealisasikan tugas akhir ini.
2. Dr. Ir. Hendra Kusuma, M.Eng.Sc., Ir. Harris Pirngadi, MT., dan Dr.Eng. Ir. Totok Mujiono, M.IKom., selaku dosen penguji atas materi tambahan yang sangat berguna untuk pengembangan Tugas Akhir ini.
3. Bapak, Ibu dan kakak yang selalu memberikan motivasi dan semangat pada penulis.
4. Dr.Eng. Ardyono Priyadi, S.T., M.Eng. selaku Ketua Jurusan Teknik Elektro ITS Surabaya.
5. Teman-teman laboratorium elektronika industri yang selalu menemani dan membantu proses pengerjaan Tugas Akhir ini.
6. Teman-teman lintas jalur elektronika 2016 yang selalu menemani dan membantu proses pengerjaan Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan dikemudian hari.

Surabaya, 26 Juni 2018

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah.....	2
1.5 Metodologi Penelitian	3
1.6 Sistematika Laporan.....	4
1.7 Relevansi.....	5
BAB II TEORI DASAR	7
2.1 Sensor IMU MPU 6050	7
2.2 Sensor Flex.....	8
2.3 Sensor MQ2	9
2.4 Sensor Posisi Potensiometer.....	11
2.5 Motor Servo	13
2.6 Arduino Uno	14
2.7 Bluetooth HC-05	15
2.8 Metode Exponential Filter.....	16
2.9 Forward Kinematika.....	16
2.10Sistem Kontrol	19
BAB III PERANCANGAN SISTEM	21
3.1 Diagram Fungsional Sistem	21

3.2 Perancangan Mekanik	23
3.2.1 Perancangan Gripper	23
3.2.2 Perancangan Lengan Robot	24
3.2.3 Perancangan Kotak Kontroler Pada Sisi Lengan Robot...	26
3.2.4 Perancangan Tempat Sensor MQ2.....	26
3.2.5 Perancangan Kotak Kontroler Pada Sisi Operator	27
3.3 Perancangan Elektrik	27
3.3.1 Perancangan Shield Arduino Uno	27
3.3.2 Perancangan Shield Arduino Nano	28
3.3.3 Perancangan Rangkaian driver Motor.....	29
3.3.4 Perancangan Rangkaian Sensor MPU6050.....	29
3.3.5 Perancangan Rangkaian Sensor Gas MQ2.....	31
3.3.6 Perancangan Rangkaian Sensor Flex	31
3.3.7 Perancangan Rangkaian Sensor posisi Potensiometer	32
3.3.8 Perancangan Rangkaian Bluetooth	33
3.4 Kontrol PID Proporsional dengan Metode Ziegler Nichols	33
3.4.1 Aturan Ziegler-Nichols	34
3.5 Perancangan Perangkat Lunak	37
3.5.1 Perangkat Lunak Sisi Operator	37
3.5.2 Perangkat Lunak Sisi Lengan Robot.....	43
BAB IV PENGUKURAN DAN PENGUJIAN	49
4.1 Pengukuran dan Pengujian Sensor MPU6050	50
4.2 Pengukuran dan Pengujian Sensor Flex	54
4.3 Pengujian Motor Servo	56
4.4 Pengukuran dan Pengujian Sensor Gas	57
4.5 Pengukuran Dan Pengujian Jarak Pengiriman Data.....	59
BAB V KESIMPULAN DAN SARAN	61
5.1 KESIMPULAN.....	61
5.2 SARAN.....	61
DAFTAR PUSTAKA	63
LAMPIRAN	65
BIODATA PENULIS	89

DAFTAR GAMBAR

HALAMAN

Gambar 2.1	Sensor MPU6050	8
Gambar 2.2	Timing Diagram MPU6050.....	8
Gambar 2.3	Diagram Dimensi Flex Sensor Series SEN 08606	9
Gambar 2.4	Rangkaian Dasar Flex Sensor.....	9
Gambar 2.5	Rangkaian Uji Sensor MQ2	10
Gambar 2.6	Karakteristik Sensitivitas Sensor MQ2	11
Gambar 2.7	Karakteristik Suhu dan Kelembaban	11
Gambar 2.8	Dimensi Potensiometer.....	12
Gambar 2.9	Rangkaian Pengganti Potensiometer	12
Gambar 2.10	Motor Servo.....	13
Gambar 2.11	Time Diagram Motor Servo	13
Gambar 2.12	Arduino Uno dan Blok Sistem Arduino	14
Gambar 2.13	Module Bluetooth.....	15
Gambar 2.14	Diagram Blok Exponential Filter	16
Gambar 2.15	Ilustrasi Kinematika Lengan 2 DOF.....	18
Gambar 2.16	Diagram Blok Kontrol.....	19
Gambar 2.17	Diagram Kontrol Close Loop	19
Gambar 3.1	Diagram Blok Sistem Sisi Operator	21
Gambar 3.2	Diagram Blok Sistem Sisi lengan robot	22
Gambar 3.3	Desain 3D Gripper	24
Gambar 3.4	Desain 2D Gripper Dengan Coreldraw	24
Gambar 3.5	Desain 3D Lengan Robot	25
Gambar 3.6	Desain 2D Lengan Robot Dengan Coreldraw	25
Gambar 3.7	Desain 2D Dengan Coreldraw.....	26
Gambar 3.8	Desain 2D Dengan Coreldraw.....	26
Gambar 3.9	Desain 2D Dengan Coreldraw.....	27
Gambar 3.10	Shield Arduino Uno	28
Gambar 3.11	Shield Arduino Nano.....	28
Gambar 3.12	Rangkaian Driver Motor Servo	29
Gambar 3.13	Rangkaian Sensor MPU6050	29
Gambar 3.14	Ilustrasi Pengambilan Sudut Pitch.....	30
Gambar 3.15	Rangkaian Sensor Gas MQ-2.....	31
Gambar 3.16	Rangkaian Sensor Flex.....	31

Gambar 3.17 Rangkaian Sensor Posisi Potensiometer	32
Gambar 3.18 Rangkaian Komunikasi Dengan Bluetooth	33
Gambar 3.19 Step Respon Metode Ziegler Nichols	34
Gambar 3.20 Grafik Respon Sistem	36
Gambar 3.21 <i>Flowchart</i> Program pada Sisi Operator.....	38
Gambar 3.22 <i>Flowchart</i> Program Sisi Robot.....	43
Gambar 4.1 Realisasi Lengan Robot.	49
Gambar 4.2 Realisasi Alat Pada Lengan Operator.....	50
Gambar 4.3 Data Yaw Sebelum Dan Sesudah Melewati Filter	51
Gambar 4.4 Data Pitch Sebelum Dan Sesudah Melewati Filter	51
Gambar 4.5 Grafik Perbandingan <i>Input</i> dan <i>Output</i> Pitch, Yaw	53
Gambar 4.6 Grafik <i>Error</i> pada Sumbu Pitch	53
Gambar 4.7 Grafik <i>Error</i> Pada Sumbu Yaw.....	54
Gambar 4.8 Data Pembacaan Sensor MQ2.....	57
Gambar 4.9 Grafik Pembacaan Nilai ADC.....	58
Gambar 4.10 Grafik Pembacaan Tegangan ADC	58

DAFTAR TABEL

HALAMAN

Tabel 3.1 Aturan Ziegler Nichols Berdasarkan Step Respon dari Plant.	35
Tabel 3.2 Nilai Parameter Ziegler-Nichols	37
Tabel 4.1 Pengujian Kesesuaian Sudut Pitch dan Yaw	52
Tabel 4.2 Data Pengujian Sensor Flex	55
Tabel 4.3 Hasil pengamatan signal motor servo	56
Tabel 4.4 Data Pengujian Sensor Gas MQ2.....	58
Tabel 4.5 Pengujian Pengiriman Data.....	59

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Beberapa tahun terakhir dunia mengalami krisis dalam keamanan dan kenyamanan dalam beraktifitas diruang publik. Hal tersebut dapat terjadi dikarenakan ada beberapa ulah kelompok tertentu yang memberikan teror berupa bom atau barang berbahaya lainnya yang diletakkan pada suatu tas yang mencurigakan kemudian diletakkan tempat-tempat umum yang ramai dengan aktifitas manusia. Sudah banyak peristiwa yang telah terjadi yang dampaknya sudah jelas sangat merugikan, dari kerugian materil hingga hilangnya nyawa manusia karena terkena ledakan bom tersebut. Terdapat 6 peristiwa teror yang besar telah terjadi di Indonesia. Yang pertama yaitu bom Bali 1 yang terjadi pada tahun 2002 yaitu rentetan tiga peristiwa pengeboman terus-menerus yang terjadi pada malam hari tanggal 12 Oktober 2002. Kedua yaitu peristiwa ledakan bom JW Marriott pada tanggal 5 Agustus 2003 di kawasan Mega kuningan Jakarta. Ketiga yaitu bom kedubes Australia pada tahun 2004 yang menelan korban jiwa sebanyak 11 orang. Keempat yaitu peristiwa bom Bali 2 yang terjadi setahun setelah bom Bali 1, teror tersebut menelan korban jiwa sebanyak 22 orang dan 102 orang luka-luka. Yang kelima yaitu peristiwa bom Hotel JW Marriot & Ritz Carlton pada tahun 2009. Kemudian yang terakhir yaitu bom Plaza Sarinah yang terjadi pada tanggal 14 Januari 2016 di sekitar Plaza Sakrinah Jl. MH Thamrin [1].

Selama ini proses inspeksi barang berbahaya yang dilakukan oleh pihak berwajib hanya dengan menggunakan anjing atau dengan mengecek barang tersebut secara langsung. Proses penginspeksian tersebut akan menimbulkan resiko yang teramat besar bagi petugas maupun anjing yang digunakan untuk menginspeksi barang yang dicurigai. Peristiwa yang telah menelan korban akibat cara tersebut yaitu AKBP Dodi Rahmawan yang kehilangan satu tangannya akibat peristiwa bom buku pada tahun 2011 di Utan kayu Jakarta Timur[2].

Dari banyaknya peristiwa teror diatas dibuatlah sebuah *prototype* lengan robot yang dapat menginspeksi barang – barang mencurigakan yang terdapat pada tempat umum. Robot ini akan bergerak sesuai dengan pergerakan lengan manusia kemudian robot ini juga dilengkapi

dengan sensor gas yang dapat memberikan informasi apakah barang tersebut menghasilkan gas berbahaya atau tidak, sehingga apabila memang benar bahan peledak maka manusia tidak akan terkena imbas ledakan dari bahan peledak tersebut.

Dari pembuatan prototipe diatas diharapkan alat ini dapat membantu pihak kepolisian dalam menangani teror – teror yang ada di dunia khususnya di Indonesia. Sehingga tidak ada lagi korban-korban dari imbas ledakan bom.

1.2 Perumusan Masalah

Pada Tugas Akhir ini yang menjadi permasalahan utama adalah sebagai berikut :

1. Bagaimana mendeteksi adanya perubahan sudut dari lengan operator.
2. Bagaimana menyesuaikan antara pergerakan lengan manusia dengan pergerakan lengan robot.
3. Bagaimana menggerakkan aktuator.
4. Bagaimana sistem kontrol yang akan digunakan.
5. Bagaimana proses pengiriman datanya

1.3 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut :

1. Dapat menentukan sudut yang terbentuk dari pergerakan lengan menggunakan sensor IMU.
2. Dapat menghasilkan sudut pergerakan yang presisi antara lengan robot dan lengan manusia.
3. Dapat menggerakkan robot menggunakan motor servo.
4. Dapat menggunakan kontrol proporsional dalam menggerakkan lengan robot.
5. Dapat mengirimkan data sensor secara *wireless*.

1.4 Batasan Masalah

Batasan masalah pada tugas akhir ini adalah:

1. Lengan robot hanya terbatas 1 dof 2 Axis ditambah 1 dof penggenggam.
2. Sensor gas MQ2 hanya mendeteksi ada atau tidak adanya gas.
3. Beban yang mampu diangkat maksimal xx Kg.

1.5 Metodologi Penelitian

Perancangan lengan robot ini terbagi menjadi delapan tahapan. Tahapan metodologi penelitian dapat dijelaskan sebagai berikut :

1. Studi Literatur Mengenai Lengan Robot.
Pada tahap ini dilakukan pembelajaran bagaimana membuat suatu lengan robot beserta perhitungan jumlah dof yang telah direncanakan.
2. Studi Literatur Mengenai Sensor IMU.
Pada tahap ini dilakukan pembelajaran mengenai karakteristik dari sensor MPU6050. Selain itu juga akan dipelajari bagaimana pengolahan *raw data* dari sensor MPU6050.
3. Studi Literatur Mengenai Sensor Flex.
Pada tahap ini dilakukan pembelajaran mengenai karakteristik dari sensor flex. Selain itu juga akan dipelajari bagaimana mengolah keluaran dari sensor flex.
4. Studi Literatur Mengenai Sensor MQ2.
Pada tahap ini dilakukan pembelajaran mengenai karakteristik dari sensor MQ2. Selain itu juga akan dipelajari bagaimana mengolah keluaran dari sensor MQ2.
5. Perancangan *Hardware* Lengan Robot.
Tahap perancangan *hardware* merupakan tahap pembuatan desain mekanik dan elektrik serta memposisikan komponen-komponen dan perangkat penyusunnya.
6. Perancangan *Software* Lengan Robot.
Tahap perancangan *software* merupakan tahap perancangan *flowchart* dari sistem yang akan dibuat. Selain itu akan dibuat juga algoritma pemrograman untuk mengendalikan lengan robot dan mengambil data dari semua sensor.
7. Pengujian Dan Perbaikan Sistem.
Pada tahap ini dilakukan pengujian terhadap alat pemindai gerak dengan menjalankan keseluruhan fungsi, dengan cara melihat hasil perbandingan perpindahan antara lengan robot dengan lengan manusia.
8. Penulisan Laporan Tugas Akhir.
Pada tahap ini dilakukan penulisan laporan tugas akhir dan juga penulisan jurnal POMITS.

1.6 Sistematika Laporan

Pembahasan tugas akhir ini dibagi menjadi lima bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Pada bab pendahuluan, menjelaskan mengenai latar belakang pemilihan topik, rumusan masalah dan batasannya. Bab ini juga membahas mengenai tujuan penelitian, metodologi, sistematika laporan, dan relevansi dari penelitian yang dilakukan.

Bab II Teori Dasar

Pada bab ini dijelaskan mengenai teori dasar dari semua sensor yang dipakai diantaranya yaitu sensor IMU, flex, dan MQ2. Selain penjelasan mengenai sensor akan dijelaskan bagaimana teori dasar dari modul bluetooth HC-05.

Bab III Perancangan Sistem

Pada bab ini dijelaskan tentang perencanaan sistem perangkat keras elektrik dan mekanik, serta perangkat lunak. Pada bab ini juga dijelaskan tentang prosedur pengujian yang dilakukan dalam penelitian.

Bab IV Pengukuran Dan Pengujian

Pada bab ini dijelaskan mengenai data hasil pengujian dari keseluruhan sistem. Pada bab ini juga dijelaskan proses analisa data yang telah didapat.

Bab V Penutup

Pada bagian bab penutup, dibahas mengenai kesimpulan dan saran dari hasil pengujian.

1.7 Relevansi

Hasil yang diharapkan dari tugas akhir ini adalah dapat menjadi alat yang bermanfaat untuk membantu pihak kepolisian dalam penginspeksian benda yang dicurigai mengandung gas berbahaya atau bahan peledak.

-----Halaman ini sengaja dikosongkan-----

BAB II

TEORI DASAR

Pada bab ini akan dibahas mengenai tinjauan pustaka dari proses pengakuisisian data dari sensor-sensor. Mulai dari mekanisme penentuan sudut yang dihasilkan dari sensor IMU MPU6050, data sensor IMU MPU6050 dapat menggerakkan motor servo, pengambilan data sensor posisi sebagai *input* feedback kontroler, pembacaan sensor gas dan mengidentifikasi jenis gas yang di tunjukkan.

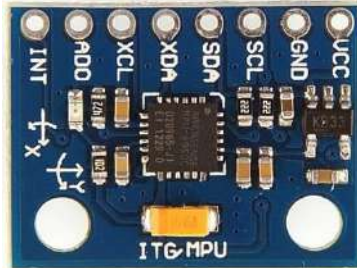
2.1 Sensor IMU MPU 6050

MPU 6050 merupakan sensor 3 axis gyroscope dan 3 axis accelero, sensor MPU dapat dilihat pada Gambar 2.4. Sensor gyroscope menggunakan prinsip dasar MEMS (*micro electro mechanical system*). Ketika gyroscope bergerak maka akan menghasilkan tegangan *output*. Pada saat gyroscope diam maka tegangan akan konstan. Ketika gyroscope berputar searah jarum jam terhadap sumbu Z maka tegangan output berkurang (- Z), dan ketika gyroscope berputar berlawanan arah jarum jam maka tegangan *output* bertambah (+ Z).

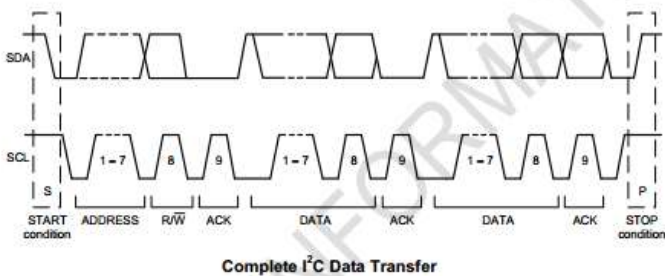
Accelerometer merupakan sebuah sensor yang menghitung percepatan maupun perubahan percepatan yang timbul akibat gravitasi bumi. Accelerometer mengukur percepatan relative terhadap frame yang popular disebut G-force. Pada saat accelero diam maka akan bernilai 1g dan ketika accelero mendeteksi adanya sebuah percepatan maka g-force akan berubah.

Sensor MPU 6050 memiliki tegangan suplay 3,3V sensor ini dapat mengirimkan data dari accelero dan gyroscope. Sensor ini berkomunikasi dengan menggunakan I2C SDA dan SCL. Untuk mengkasesk data dari MPU 6050 ini maka perlu dilakukannya pengaturan pada pengambilan data melalui I2C. bentuk fisik MPU 6050 dapat dilihat pada Gambar 2.1.

Untuk mendapatkan data gyro dan accelero dari MPU6050 maka kita harus mengatur pin SDA SCL untuk mengikuti diagram yang ada pada Gambar 2.2. Pertama pin SDA dan SCL diset HIGH kemudian dengan mengakses alamat dari gyro dan accelero. Jika sudah mengikuti diagram diatas maka kita akan mendapatkan data dari pembacaan sensor gyro dan accelero.



Gambar 2.1 Sensor MPU6050 [17].

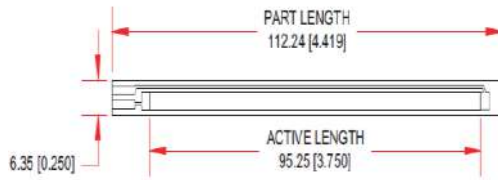


Gambar 2.2 Timing Diagram MPU6050 [17].

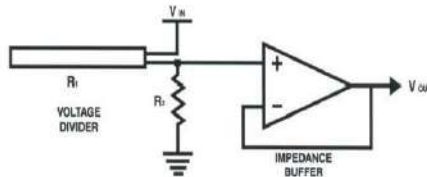
2.2 Sensor Flex

Flex Sensor merupakan sebuah sensor fleksibel yang memiliki panjang 4,5 inch. Sensor flex menghasilkan suatu resistansi apabila ditekuk. Nilai resistansi dari sensor ini berubah – ubah seiring seberapa besar nilai sudut yang di bentuk. Berikut adalah spesifikasi dari sensor flex :

1. Cangkupan suhu : -35°C sampai +80°C
2. Hambatan datar : 10K Ohm
3. Toleransi hambatan : ±30%
4. Cakupan hambatan tekukan : 60K Ohm
5. Nilai power : 0,5 Watt dst. 1 Watt sampai batas maksimal.



Gambar 2.3 Diagram Dimensi Flex Sensor Series SEN 08606 [15].



Gambar 2.4 Rangkaian Dasar Flex Sensor [1].

Berikut pada Gambar 2.3 adalah diagram dimensi sebuah sensor flex, dan pada Gambar 2.4 adalah rangkaian dasar dari sensor flex.

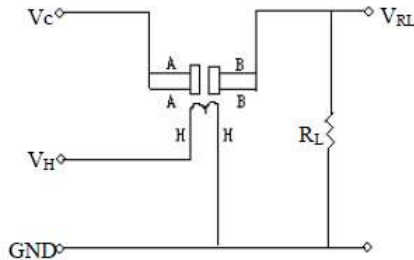
Tegangan *output* dari sensor dapat dirumuskan seperti pada persamaan 2.1.

$$V_{out} = \frac{R_2}{R_1 + R_2} \times V_{in} \quad (2.1)$$

2.3 Sensor MQ2

MQ-2 adalah sensor semikonduktor untuk gas yang mudah terbakar. Bahan sensitif dari sensor gas MQ-2 adalah SnO₂, yang dengan konduktivitas rendah di udara bersih. Ketika target gas yang mudah terbakar ada. Konduktivitas sensor lebih tinggi seiring dengan meningkatnya konsentrasi gas. Sensor gas MQ-2 memiliki kepekaan yang tinggi terhadap LPG, Propana dan Hidrogen. Selain itu sensor MQ2 juga dapat digunakan untuk metana dan uap yang mudah terbakar lainnya.

Terdapat beberapa kegunaan dari sensor MQ2 diantaranya yaitu detektor kebocoran gas domestik, detektor gas industri yang mudah terbakar dan detektor gas portabel.

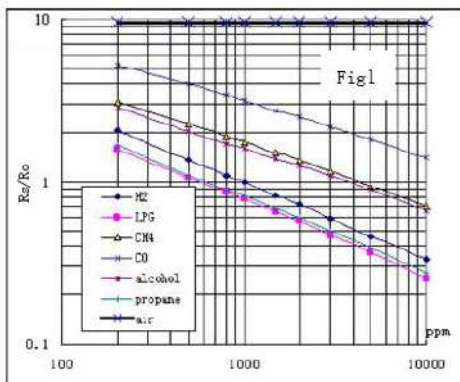


Gambar 2.5 Rangkaian Uji Sensor MQ2 [16].

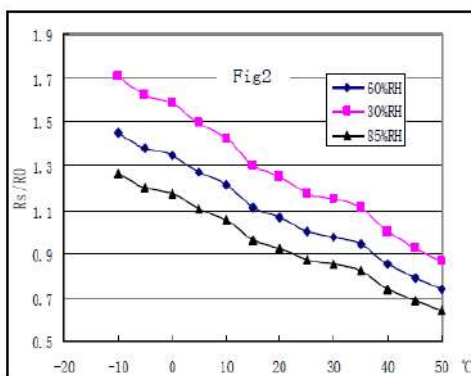
Pada Gambar 2.5 ditunjukkan rangkaian uji dasar dari sensor.. Sensor harus diletakkan 2 tegangan, tegangan pemanas (VH) dan tegangan uji (VC). VH digunakan untuk memasok suhu kerja ke sensor, sementara VC digunakan untuk mendeteksi tegangan (VRL) pada resistansi beban (RL) yang secara seri dengan sensor. Sensor memiliki polaritas cahaya, Vc membutuhkan daya DC. VC dan VH dapat menggunakan rangkaian daya yang sama dengan prakondisi untuk menjamin kinerja sensor. Untuk membuat sensor dengan kinerja yang lebih baik, nilai RL yang sesuai dibutuhkan: Kekuatan Sensitivitas (Ps):

$$P_s = \frac{V_{c2} \times R_s}{(R_s + R_L) \times 2} \quad (2.2)$$

Pada Gambar 2.6 menunjukkan karakteristik sensitivitas khas MQ-2, ordinat berarti rasio resistensi dari sensor (R_s / R_o), absis adalah konsentrasi gas. R_s berarti resistensi dalam gas yang berbeda, R_o berarti resistensi sensor dalam 1000ppm Hyrogen. Semua tes berada dalam kondisi uji standar.



Gambar 2.6 Karakteristik Sensitivitas Sensor MQ2 [16].



Gambar 2.7 Karakteristik Suhu dan Kelembaban [16].

Pada Gambar 2.7 menunjukkan karakteristik suhu dan kelembaban yang khas. Ordinat berarti rasio resistensi dari sensor (R_s / R_o), R_s berarti resistansi sensor dalam 1000ppm. R_o berarti ketahanan sensor di lingkungan 1000ppm metana.

2.4 Sensor Posisi Potensiometer

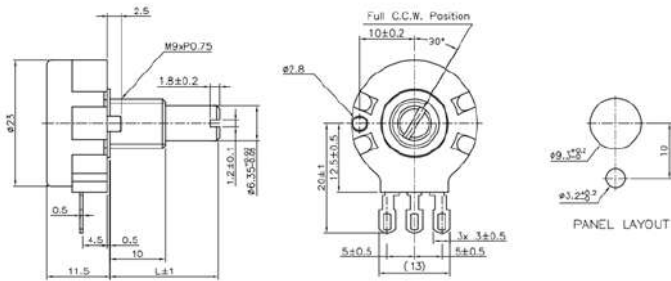
Potensiometer adalah resistor tiga terminal dengan sambungan geser yang membentuk pembagi tegangan dapat diatur. Jika hanya dua terminal yang digunakan (salah satu terminal tetap dan terminal geser),

potensiometer berperan sebagai resistor variabel atau Rheostat. Potensiometer biasanya digunakan untuk mengendalikan piranti elektronik seperti pengendali suara pada penguat. Potensiometer yang dioperasikan oleh suatu mekanisme dapat digunakan sebagai transduser. Dimensi dari suatu potensio dapat dilihat pada Gambar 2.8.

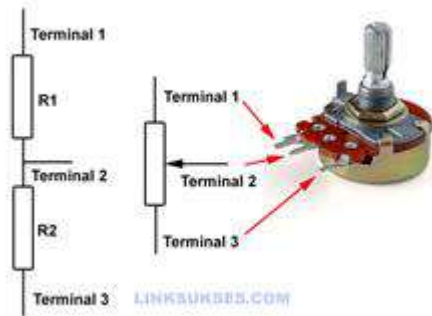
Output potensiometer ini berupa resistansi yang linier sehingga dapat digunakan untuk mengukur perubahan posisi dari perubahan resistansi. *Output* dari potensiometer dapat ditunjukkan pada persamaan berikut :

$$V_{out} = \frac{R_1}{R_1 + R_2} \times V_{in} \quad (2.3)$$

Rangkaian pengganti dari suatu potensiometer ditunjukkan pada Gambar 2.8



Gambar 2.8 Dimensi Potensiometer [18].



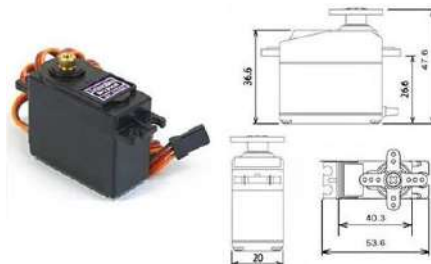
Gambar 2.9 Rangkaian Pengganti Potensiometer [18].

2.5 Motor Servo

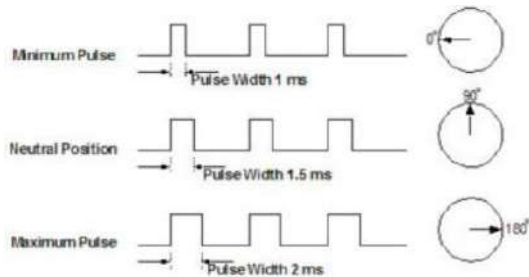
Motor servo adalah jenis motor DC, yang membedakan motor servo dengan motor DC biasa adalah motor servo dapat diatur perputaran sudutnya. Ada 2 jenis motor servo yang pertama adalah motor servo standart dan continuous. Motor servo standart dapat berputar 180° sedangkan motor servo countinuous adalah motor servo dengan perputaran sampai 360° .

Motor servo memiliki 3 pin *input*, yaitu Vcc, Gnd, dan Data. Vcc dan Gnd merupakan sumber tegangan untuk motor servo, sedangkan data merupakan data dari mikrokontroler yang berupa Pulse Widht Modulation (PWM). PWM merupakan teknik modulasi yang merubah panjang sinyal high atau low dalam frekuensi yang tetap. Bentuk dari motor servo dapat dilihat pada Gambar 2.10.

Motor servo memiliki panjang pulsa 20ms, untuk mendapat sudut 0° maka pulsa high dapat diset selama 1 ms, dan untuk mendapat sudut 180° maka lama waktu high diatur selama 2ms dan waktu *low* adalah sisa dari periode 20ms tersebut. Time diagram pada servo dapat dilihat pada Gambar 2.11.

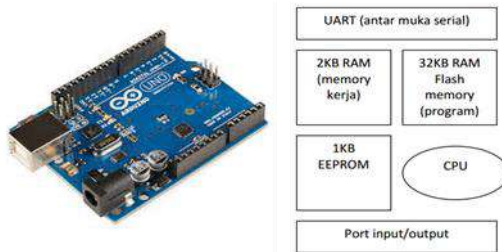


Gambar 2.10 Motor Servo [19].



Gambar 2.11Time Diagram Motor Servo [19].

2.6 Arduino Uno



Gambar 2.12 Arduino Uno dan Blok Sistem Arduino

Dalam pembuatan tugas akhir, mikrokontroler merupakan sebuah komponen utama. Mikrokontroler terdiri dari CPU, Memory, I/O, dan perangkat penunjang lain. Mikrokontroler merupakan salah satu komponen penting dalam sistem ini, mikrokontrolerlah yang bertindak dalam proses pengolahan sinyal, dan proses kontrol *hardware*.

Pada tugas akhir ini mikrokontroler yang digunakan yaitu Arduino Uno. Arduino Uno adalah kombinasi dari perangkat keras, bahasa pemrograman dan *Integrated Development Environment* (IDE) yang canggih.

Arduino Uno merupakan papan mikrokontroler berbasis ATmega328 yang mempunyai 14 pin masukan dan keluaran digital (termasuk 6 diantaranya dapat berfungsi sebagai keluaran Pulse Width Modulation), 8 masukan analog, sebuah kristal 16 MHz, dan sebuah tombol reset. Arduino Uno beroperasi pada tegangan 5 volt.

Berikut adalah spesifikasi dari Arduino Uno :

1. Tegangan Operasi : 5V
2. Tegangan *Input* : 7 - 12V
3. Batas Tegangan *Input* : 6 - 20V
4. Pin Digital I/O : 14 Pin Analog Inpu 6
5. Arus DC per I/O Pin : 40 mA
6. Arus DC untuk pin : 3.3V 50 mA
7. Flash Memory : 32 KB (ATmega328)
8. SRAM : 2 KB (Atmega328)
9. EEPROM : 1 KB (Atmega328)
10. Clock : 16 MHz

2.7 Bluetooth HC-05

HC-05 Adalah sebuah modul Bluetooth SPP (*Serial Port Protocol*) yang mudah digunakan untuk komunikasi serial *wireless* (nirkabel) yang mengkonversi port serial ke Bluetooth. HC-05 menggunakan modulasi bluetooth V2.0 + EDR (*Enhanced Data Rate*) 3 Mbps dengan memanfaatkan gelombang radio berfrekuensi 2,4 GHz.

Pada tugas akhir ini bluetooth digunakan sebagai komunikasi antara 2 bluetooth. Modul ini dapat digunakan sebagai slave maupun master. HC-05 memiliki 2 mode konfigurasi, yaitu AT mode dan *Communication mode*. AT mode berfungsi untuk melakukan pengaturan konfigurasi dari HC-05. Sedangkan *Communication mode* berfungsi untuk melakukan komunikasi bluetooth dengan piranti lain. Dalam penggunaannya, HC-05 dapat beroperasi tanpa menggunakan driver khusus. Untuk berkomunikasi antar Bluetooth, minimal harus memenuhi dua kondisi berikut :

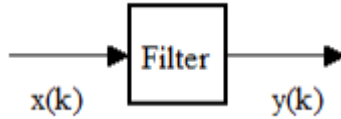
1. Komunikasi harus antara master dan slave.
2. Password harus benar (saat melakukan *pairing*).

Jarak sinyal dari HC-05 adalah 30 meter, dengan kondisi tanpa halangan. Bentuk fisik dari modul Bluetooth dapat dilihat pada Gambar 2.13.



Gambar 2.13 Module Bluetooth [14].

2.8 Metode Exponential Filter



Gambar 2.14 Diagram Blok Exponential Filter

Filter eksponensial adalah filter lolos rendah, yang dimaksudkan untuk melewatkan frekuensi rendah dan menipiskan frekuensi tinggi. Metode ini hanya memiliki satu parameter metode pengaturan. Metode ini membutuhkan penyimpanan hanya satu variabel keluaran sebelumnya. Diagram blok dari filter exponential dapat dilihat pada Gambar 2.14.

Filter eksponensial adalah kombinasi dari perkiraan sebelumnya (keluaran) dengan data masukan terbaru, dengan jumlah bobot sama dengan 1 sehingga keluaran sesuai dengan masukan pada kondisi tunak. Berikut persamaan filter eksponensial :

$$Y(k) = a * Y(k - 1) + (1 - a) * x(k) \quad (2.4)$$

Dimana :

$X(k)$ = Raw *input* saat waktu step k

$Y(k)$ = *Output* terfilter saat waktu step k

A adalah konstanta antara 0 sampai 1, normalnya antara 0.8 dan 0.99. $(a-1)$ atau a terkadang dikenali dengan istilah “*smoothing constan*”

2.9 Forward Kinematika

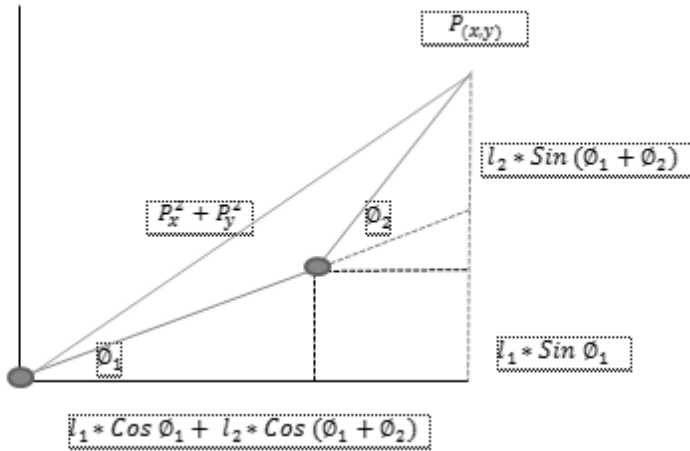
Untuk melakukan proses pemetaan antara gerakan robot dan gerakan manusia secara garis besar terdapat dua cara yang berbeda untuk melakukannya. Cara pertama adalah robot melakukan motion primitives yang telah dibuat sebelumnya saat software imitasi mendeteksi gerakan manusia yang menginterpretasikan gerakan yang berkorespondensi dengan motion primitives yang ada. Motion primitives

ini terlebih dulu harus diprogramkan ke robot menggunakan metode inverse kinematik atau perangkat lunak lain yang dapat membuat program. Setelah menyimpan gerakan-gerakan yang diinginkan, perangkat lunak penangkap citra diprogramkan sedemikian rupa untuk menentukan motion primitives mana yang akan dilakukan robot dari citra yang ditangkap oleh perangkat capture. Masalah dari teknik ini adalah gerakan yang dilakukan robot dapat sangat berbeda dengan gerakan aktor pendemonstrasi. Agar dapat melakukan imitasi yang baik harus dibuat database yang besar dari motion primitives, dan perancangan perangkat lunak yang baik, yang dapat membedakan gerakan-gerakan yang dilakukan oleh aktor. Masalah lain pada teknik ini sulitnya mendapatkan transisi antar gerakan yang halus.

Teknik lain yang digunakan untuk melakukan pemetaan adalah dengan menggunakan metode inverse kinematik. Kinematika adalah bagian dari ilmu fisika yang membahas pergerakan robot dengan mengabaikan gaya pada objek tersebut. Dengan kinematika, telah dikembangkan pengetahuan lebih lanjut tentang posisi, kecepatan, percepatan, dan derivative dari posisi dengan orde yang lebih tinggi. Umumnya, robot terdiri dari link-link yang dihubungkan dengan joint (sendi) yang memungkinkan pergerakan terhadap link yang saling bertentangan.

Dalam kinematika dikenal istilah forward kinematik dan inverse kinematik. Masalah yang terdapat pada forward kinematik adalah, “Dimana posisi end effector apabila diberikan semua sudut pada setiap joint robot ($x=f(\theta)$)” Sedangkan inverse kinematik mencoba menjawab pertanyaan “Berapa nilai sudut pada setiap joint robot apabila diberikan nilai posisi yang diinginkan ($\theta = f(x)$)” Ketika manusia menggerakkan tangan menuju suatu posisi tertentu, secara tidak langsung manusia melakukan penyelesaian masalah inverse kinematik. Metode ini memetakan titik-titik yang didapat dari objek tracking menjadi sudut pada manipulator. Metode ini cukup sulit dilakukan apabila jumlah joint robot kurang dari jumlah joint pada aktor.

Contoh permasalahan forward kinematik dan inverse kinematik adalah pada model robot serial link planar dua sendi seperti pada berikut.



Gambar 2.15 Ilustrasi Kinematika Lengan 2 DOF

Dalam *Inverse* Kinematik 2DoF memiliki kesulitan jika diselesaikan dengan menggunakan persamaan trigonometri, tetapi masih dapat diselesaikan dengan cara berikut. Berdasarkan dari *forward* kinematik 2 Dof didapatkan persamaan seperti yang ditunjukkan pada persamaan 2.5 dan 2.6 :[3],[9]

$$P_x = l_1 * \cos \phi_1 + l_2 * \cos (\phi_1 + \phi_2) \quad (2.5)$$

$$P_y = l_1 * \sin \phi_1 + l_2 * \sin (\phi_1 + \phi_2) \quad (2.6)$$

Sedangkan nilai teta 1 dan teta 2 dari serial link diatas dapat memiliki solusi inverse kinematik sebagai berikut:

:

$$\cos \phi_2 = \frac{P_x^2 + P_y^2 - l_1^2 - l_2^2}{2l_1 l_2} \quad (2.7)$$

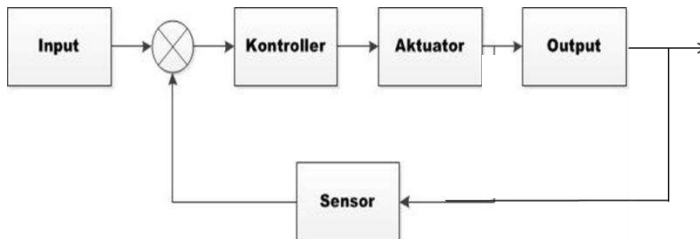
Untuk teta 1 didapatkan persamaan sebagai berikut

$$\phi_1 = \text{atan2}^{-1}(P_x, P_y) - \text{atan2}^{-1}(k_2, k_1) \quad (2.8)$$

2.10 Sistem Kontrol



Gambar 2.16 Diagram Blok Kontrol.



Gambar 2.17 Diagram Kontrol Close Loop.

Sistem kontrol (*control system*) adalah suatu alat untuk mengendalikan, memerintah, dan mengatur keadaan dari suatu sistem. Sasaran sistem kontrol adalah untuk mengatur keluaran (*output*) dalam kondisi atau keadaan yang telah ditetapkan oleh masukan (*input*) melalui elemen sistem kontrol. Pada Gambar 2.16 memperlihatkan suatu diagram umum suatu sistem kontrol. Sistem kontrol terbagi dua, yaitu sistem kontrol dengan *openloop* dan sistem kontrol *closeloop*. Pada pembuatan tugas akhir ini sistem kontrol yang digunakan adalah sistem kontrol dengan *closeloop*. Sistem kontrol *closeloop* identik dengan sistem kontrol umpan balik, dimana nilai dari keluaran akan ikut mempengaruhi pada aksi kontrolnya. Secara umum, sistem kontrol *closeloop* diberikan oleh Gambar 2.17. Gambar 2.17 menunjukkan hubungan masukan dan keluaran dari sistem kontrol *closeloop*. Jika dalam hal ini manusia bekerja sebagai operator, maka manusia ini akan menjaga sistem agar tetap pada keadaan yang diinginkan. Ketika terjadi perubahan pada sistem maka manusia akan melakukan langkah-langkah awal pengaturan sehingga sistem kembali bekerja pada keadaan yang diinginkan. Sistem *closeloop* ini digunakan untuk mengatur posisi dari motor servo yang ada pada lengan robot. Sehingga dengan menggunakan sistem *closeloop* diharapkan *output* dapat tetap terjaga pada *setpoint* yang diberikan.

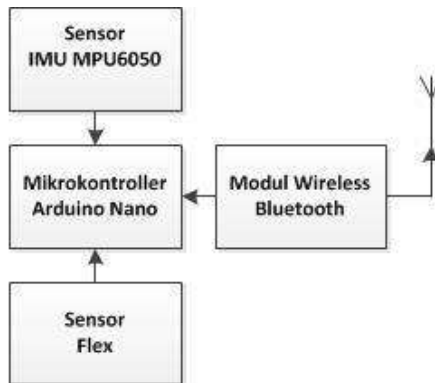
-----Halaman ini sengaja dikosongkan-----

BAB III PERANCANGAN SISTEM

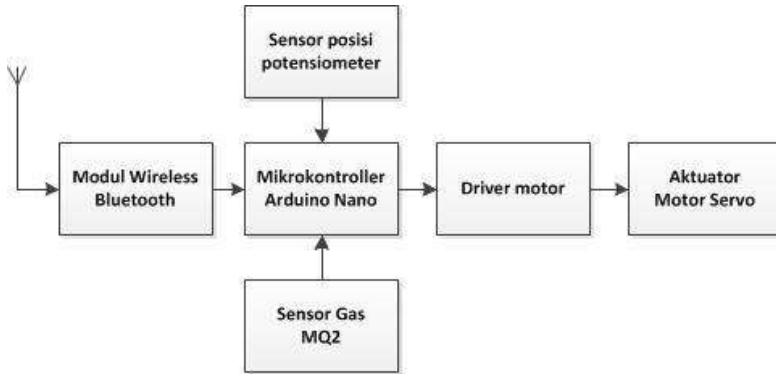
Pada bab ini berisi tahapan mengenai tahapan yang dilakukan dalam perencanaan dan pembuatan Tugas Akhir. Penjelasan diawali dengan blok fungsional sistem secara keseluruhan yang meliputi proses kerja alat dalam bentuk alur diagram. Perancangan mekanik yang membahas tentang desain dan pembuatan mekanik yang mendukung cara kerja alat. Perancangan elektrik yang membahas perancangan rangkaian elektrik sebagai rangkaian pendukung alat. Serta perancangan perangkat lunak yang meliputi perancangan diagram alur program serta pengaplikasian kontrol proporsional untuk mengendalikan motor servo.

3.1 Diagram Fungsional Sistem

Sebelum melakukan perancangan perangkat keras dan perangkat lunak, diperlukan sebuah perancangan blok fungsional sistem berupa blok diagram yang menjelaskan sistem kerja secara keseluruhan Tugas Akhir ini. Secara keseluruhan blok fungsional sistem sisi operator dan sisi lengan robot dapat dilihat pada Gambar 3.1 dan 3.2.



Gambar 3.1 Diagram Blok Sistem Sisi Operator.



Gambar 3.2 Diagram Blok Sistem Sisi lengan robot.

Sesuai dengan Gambar 3.1 dan 3.2 , dijelaskan tentang sistem kerja dari kendali lengan robot yang meniru pergerakan tangan untuk inspeksi objek yang mengandung gas berbahaya. Berikut adalah penjelasan blok diagram pada sisi lengan robot,

Pada Gambar 3.1 berawal dari pembacaan sensor IMU MPU6050 dengan cara pembacaan *raw data* dari sensor berupa nilai *accelerometer* sumbu x, y dan z pada alamat *register* 0x3B sampai 0x40. Kemudian pembacaan nilai *gyrometer* sumbu x,y dan z pada alamat *register* 0x43 sampai 0x48. Dari *raw data* yang telah diperoleh akan dikonversi menjadi sudut dengan persamaan 3.1 untuk pitch dan persamaan 3.3 untuk yaw.

$$\theta = \tan^{-1} \left(\frac{AcY}{AcZ} \right) \times \frac{180}{3.14} \quad (3.1)$$

$$raw_to_dps = \frac{GyZ}{65.356} \quad (3.2)$$

$$sudut_yaw = sudut_yaw + raw_to_dps/250 + 0.00404 \quad (3.3)$$

Dari data tersebut diambil data pitch dan yaw saja dikarenakan pergerakan lengan robot hanya menggunakan 2 axis. *Mapping* dari sensor menuju lengan robot yaitu 0° sampai 180 °. Data yang dikirim oleh sensor menuju *actuator* lengan robot sudah berupa sudut sehingga pada motor servo apabila mendapat *input* sudut, maka akan menggerakkan dengan nilai PWM tertentu sesuai sudut yang diberikan.

Kemudian dilakukan pembacaan nilai dari sensor flex. Data dari sensor flex ini akan digunakan untuk menggerakkan gripper dari lengan robot. Setelah kedua data dari sensor di dapatkan maka tugas dari

mikro kontroler arduino nano yaitu mengolah data tersebut. Pengolahan data dari sensor IMU digunakan exponential filter supaya didapatkan data yang lebih stabil apabila terjadi perubahan posisi dari sensor IMU. Kemudian untuk data dari sensor flex diolah supaya didapatkan perubahan sudut dari sensor menghasilkan perubahan nilai ADC yang linier. Komunikasi antara 2 modul bluetooth ini memerlukan pengaturan tersendiri supaya antara dua modul bluetooth dapat terhubung secara otomatis dan saling bertukar informasi. Pengaturan bluetooth menggunakan perintah AT-Commnad. Setelah data dari kedua sensor diolah dan modul bluetooth telah siap kemudian data tersebut akan dikirimkan menggunakan bluetooth menuju Slave.

Pada Gambar 3.2 dijelaskan diagram blok untuk sisi slave. Pada sisi slave dijelaskan bahwasannya data yang diterima dari master akan menjadi nilai setpoint untuk menjalankan aktuator. Aktuator yang digunakan yaitu 3 motor servo. Dari kelompok data yang diterima dilakukan parsing data untuk memisahkan data antara sensor flex dan sensor MPU6050. Untuk sensor MPU6050 sendiri dilakukan pemisahan data antara data sudut yaw dan pitch.

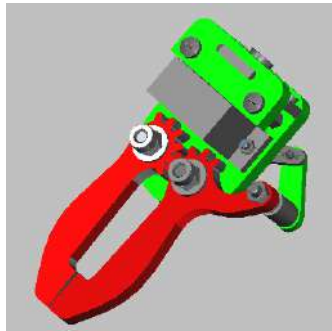
Pada mikrokontroler arduino slave ini selain menjalankan perintah menerima data dari master dan menjalankan aktuator, mikrokontroler ini juga melakukan pengambilan data dari sensor gas MQ2 dan pengambilan data dari sensor posisi Potensiometer yang digunakan sebagai feedback kontrol posisi dari pergerakan motor servo. Pengendalian motor servo menggunakan sistim kontrol proporsional. Sehingga diharapkan dapat menghasilkan pergerakan servo yang sesuai dengan setpoint.

3.2 Perancangan Mekanik

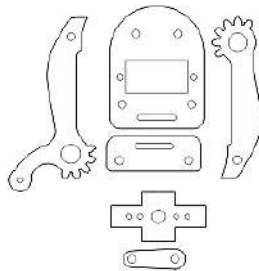
Pada sub bab ini akan dibahas mengenai perancangan mekanik untuk tugas akhir ini. Perancangan mekanik berupa perancangan perangkat keras yang mendukung seluruh perancangan dan pembuatan alat. Perancangan mekanik yang akan dibahas meliputi perancangan lengan robot, gear untuk sesnor posisi, tempat peletakan sensor gas dan tempat mikrokontroler sebagai tempat rangkaian elektrik.

3.2.1 Perancangan Gripper

Pada perancangan gripper atau penggenggam ini direncanakan mampu bergerak 1 dof dengan lebar sudut yang dapat dibuat 60 derajat. Dimensi dari gripper yaitu 10 cm x 3 cm x 2,5cm



Gambar 3.3 Desain 3D Gripper

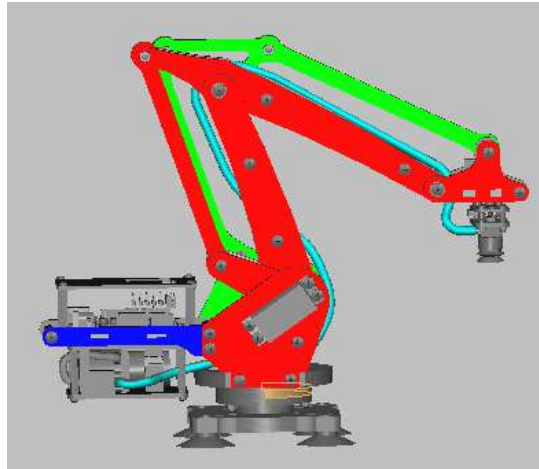


Gambar 3.4 Desain 2D Gripper Dengan Coreldraw

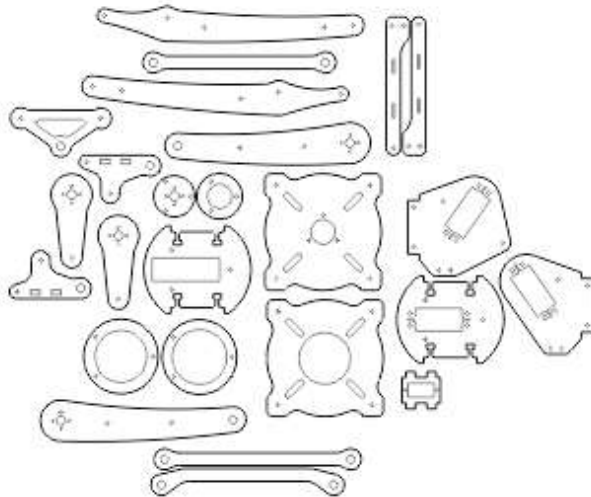
Pada perancangan gripper ini digunakan bahan berupa akrilik dengan ukuran tebal 4mm. Desain 3D dari *gripper* dapat dilihat pada Gambar 3.3, Sedangkan Desain 2D untuk pembuatan akrilik dapat dilihat pada Gambar 3.4.

3.2.2 Perancangan Lengan Robot

Pada perancangan lengan robot ini digunakan bahan berupa akrilik dengan ukuran tebal 4mm. Lengan robot direncanakan mampu bergerak 2dof dan 2 Axis. Lengan robot dibuat berbahan dasar akrilik dengan tebal 4mm. Dengan jangkauan lengan maksimal 60cm. Dimensi dari lengan robot yaitu 30cm x 30cm x 45 cm. Desain 3D dari lengan robot dapat dilihat pada Gambar 3.5, sedangkan desain 2D dapat dilihat pada Gambar 3.6.



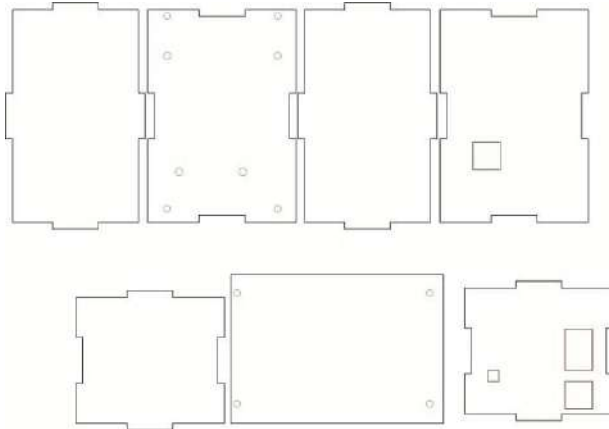
Gambar 3.5 Desain 3D Lengan Robot



Gambar 3.6 Desain 2D Lengan Robot Dengan Coreldraw

3.2.3 Perancangan Kotak Kontroler Pada Sisi Lengan Robot

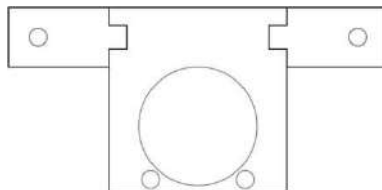
Pada perancangan kotak kontroler pada sisi slave ini digunakan Akrilik dengan tebal 3mm. Proses pembuatan desain menggunakan *software* coreldraw. Hasil dari desain menggunakan coreldraw dapat dilihat pada Gambar 3.7.



Gambar 3.7 Desain 2D Dengan Coreldraw

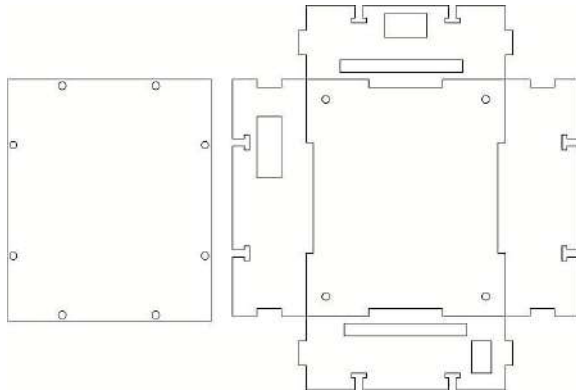
3.2.4 Perancangan Tempat Sensor MQ2

Pada perancangan tempat sensor gas MQ2 ini digunakan Akrilik dengan tebal 3mm. Proses pembuatan desain menggunakan *software* coreldraw. Hasil dari desain menggunakan coreldraw dapat dilihat pada Gambar 3.8.



Gambar 3.8 Desain 2D Dengan Coreldraw

3.2.5 Perancangan Kotak Kontroler Pada Sisi Operator



Gambar 3.9 Desain 2D Dengan Coreldraw

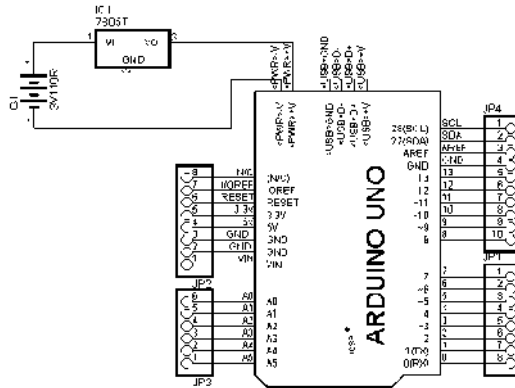
Pada perancangan kotak kontroler pada sisi Operator ini digunakan Akrilik dengan tebal 3mm. Proses pembuatan desain menggunakan *software* coreldraw. Hasil dari desain menggunakan coreldraw dapat dilihat pada Gambar 3.9.

3.3 Perancangan Elektrik

Pada sub bab ini akan dibahas mengenai perancangan eletrik untuk tugas akhir ini. Perancangan elektrik berupa perancangan perangkat keras yang mendukung seluruh perancangan dan pembuatan alat. Perancangan mekanik yang akan dibahas meliputi perancangan shield sistem minimum arduino uno dan nano, rangkaian sensor gas MQ2, sensor MPU6050, sensor flex dan sensor posisi potensiometer.

3.3.1 Perancangan Shield Arduino Uno

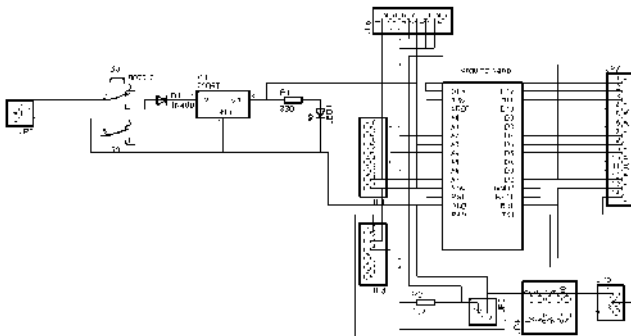
Pada perancangan shield arduino ini digunakan untuk mempermudah pemasangan modul sensor yang akan digunakan. Shield arduino uno ini diletakkan pada sisi lengan robot. Shield ini terdiri dari beberapa komponen penyusun diantaranya yaitu rangkaian pinout expansion dan rangkaian power. Rangkaian power ini menggunakan LM7805 untuk mensupplay tegangan sebesar 5V. Perancangan shield arduino uno dapat dilihat pada Gambar 3.10.



Gambar 3.10 Shield Arduino Uno

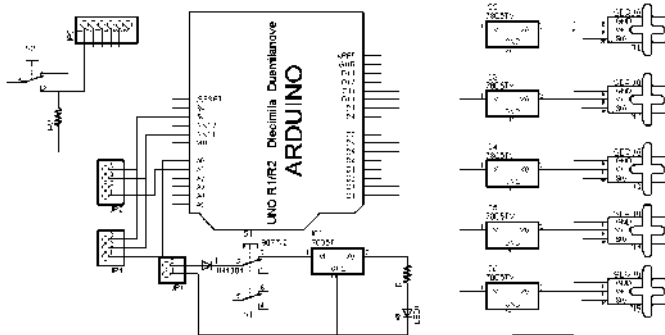
3.3.2 Perancangan Shield Arduino Nano

Pada perancangan shield arduino nano ini digunakan untuk mempermudah pemasangan modul sensor yang akan digunakan. Shield arduino nano ini diletakkan pada sisi lengan operator. Shield ini terdiri dari beberapa komponen penyusun diantaranya yaitu rangkaian pinout expansion, rangkaian sensor dan rangkaian power. Rangkaian power ini menggunakan LM7805 untuk mensupplay tegangan sebesar 5V. Sensor yang berada pada shield ini yaitu berupa sensor Flex dan sensor MPU6050. Perancangan shield arduino nano dapat dilihat pada Gambar 3.11.



Gambar 3.11 Shield Arduino Nano

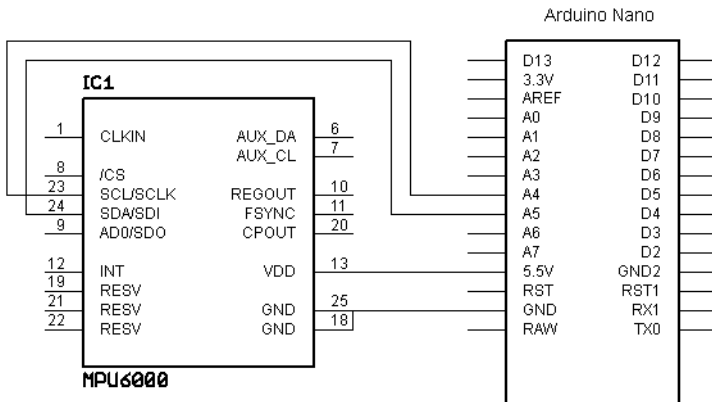
3.3.3 Perancangan Rangkaian driver Motor



Gambar 3.12 Rangkaian Driver Motor Servo

Rangkaian driver motor servo disini menggunakan LM7805 untuk mensupply tiap motor servo. Pada tugas akhir ini digunakan 3 motor servo sehingga digunakan 3 LM7805 untuk memberi tegangan pada Servo. Kemudian untuk pin *input* sinyal dari motor servo dihubungkan secara langsung pada pin PWM arduino Uno, pin PWM yang digunakan yaitu pin 9, 10 dan pin 11. Motor Servo yang digunakan pada tugas akhir ini bertipe MG995. Rangkaian Driver motor servo dapat dilihat pada Gambar 3.12.

3.3.4 Perancangan Rangkaian Sensor MPU6050



Gambar 3.13 Rangkaian Sensor MPU6050

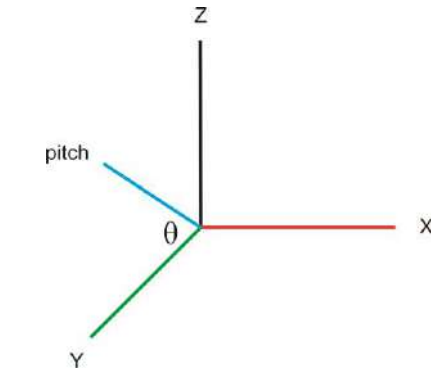
Pada perancangan rangkaian sensor MPU6050, pinout dari MPU6050 yang digunakan yaitu pin SDA, SCL, INT, VCC dan GND. Sedangkan pada sisi arduino nano digunakan pin A4 untuk SDA, A5 untuk SCL, D2 untuk INT kemudian VCC dan GND. Perancangan rangkaian MPU6050 dapat dilihat pada Gambar 3.13.

Accelerometer mungkin tampak tidak terkait dengan orientasi perangkat, karena mengukur tiga percepatan linear objek. Namun, percepatan diukur relatif terhadap jatuh bebas, yang berarti bahwa ketika perangkat tetap diam, harus ada percepatan besarnya $\sqrt{a_x^2 + a_y^2 + a_z^2} = 9.8 \text{ m/s}^2$. Cara besarnya ini didistribusikan pada tiga sumbu percepatan tetap (t), $a_y(t)$ dan $a_z(t)$ akan memberi kita informasi tentang orientasi absolut. AcY adalah nilai yang didapat dari pembacaan raw data dari sensor MPU6050. AcZ adalah nilai yang didapat dari pembacaan raw data dari sensor MPU6050.

Berdasarkan Gambar 3.14 dapat diambil nilai sudut teta dengan persamaan

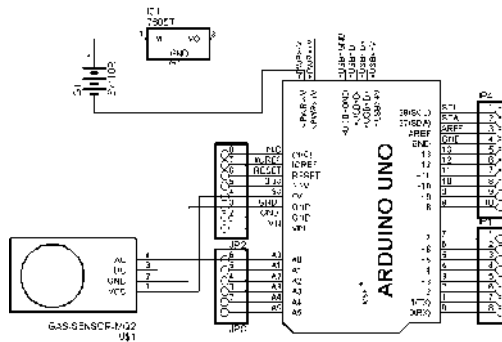
$$\theta = \tan^{-1} \left(\frac{AcY}{AcZ} \right) \times \frac{180}{3.14} \quad (3.4)$$

Perhitungan yang sama dilakukan pada MPU6050 yang kedua. Setelah didapatkan hasil konversi sudut dari kedua MPU6050 maka salah satu MPU dijadikan referensi sehingga untuk mendapatkan sudut yang teliti nilai perubahan sudut MPU pertama dikurangi dengan pembacaan sudut MPU kedua.



Gambar 3.14 Ilustrasi Pengambilan Sudut Pitch

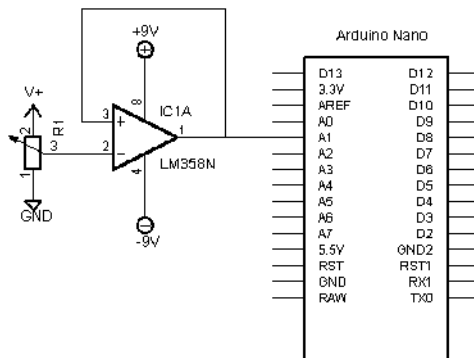
3.3.5 Perancangan Rangkaian Sensor Gas MQ2



Gambar 3.15 Rangkaian Sensor Gas MQ-2

Pada sensor gas MQ2 terdapat 4 macam pin *output*, diantaranya yaitu pin VCC, GND, D0, dan A0. Untuk pin D0 yaitu ketika nanti dihubungkan pada pin tersebut maka *output* bernilai digital sedangkan A0 merupakan pin *output* yang apabila dihubungkan pada pin tersebut maka *output* bernilai Analog. Pada tugas akhir ini digunakan pin A0 karena pada *output* A0 akan diambil nilai konversi ADC oleh Arduino Uno pada pin A0. Rangkaian sensor gas MQ2 dapat dilihat pada Gambar 3.15.

3.3.6 Perancangan Rangkaian Sensor Flex



Gambar 3.16 Rangkaian Sensor Flex

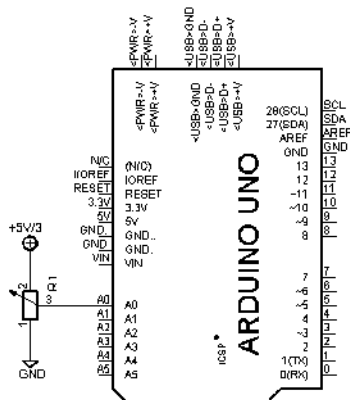
Output dari sensor flex ini berupa resistansi maka dari itu digunakanlah resistor untuk mengkonversi menjadi tegangan sehingga nantinya dapat dibaca oleh ADC dari arduino nano. Penambahan komponen berupa op amp digunakan apabila terjadi tegangan drop yang diakibatkan karena terlalu panjang kabel dari sensor menuju mikrokontroler. op amp yang digunakan LM358 dengan konfigurasi noninverting amplifier. Pin yang digunakan arduino nano untuk membaca nilai ADC yaitu pada pin A1. Rangkaian sensor flex dapat dilihat pada Gambar 3.16.

Untuk mendapatkan nilai konversi dari resistansi menjadi tegangan dapat menggunakan persamaan 3.2

$$V_{out} = V_{input} \left(\frac{R_1}{R_1 + R_2} \right) \quad (3.5)$$

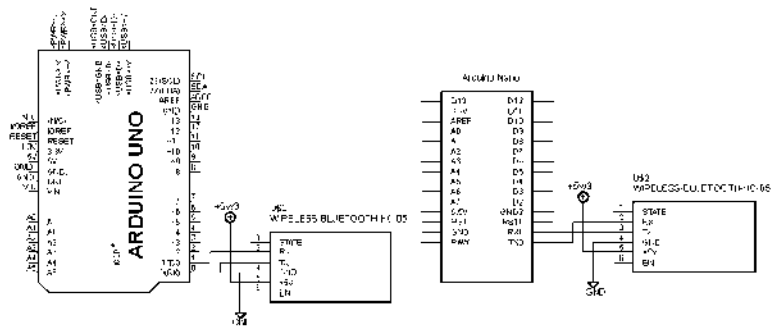
3.3.7 Perancangan Rangkaian Sensor posisi Potensiometer

Pada perancangan sensor posisi yang menggunakan potensiometer ini pin variable dari potensiometer dihubungkan pada pin A2 arduino uno. Dari perubahan nilai resistansi dari potensiometer dapat diukur nilai konversi ADC nya, sehingga dari nilai ADC tersebut dapat dihitung nilai posisi sudut dari perputaran potensiometer. Rangkaian dari sensor posisi ini dapat dilihat pada Gambar 3.17.



Gambar 3.17 Rangkaian Sensor Posisi Potensiometer

3.3.8 Perancangan Rangkaian Bluetooth



Gambar 3.18 Rangkaian Komunikasi Dengan Bluetooth

Pada perancangan rangkaian elektrik dari tugas akhir ini menggunakan 2 modul bluetooth yang difungsikan sebagai master dan slave. Pin yang digunakan yaitu Tx, Rx, VCC, dan GND. Konfigurasi pin Tx dan Rx ini dihubungkan secara silang pada pin Arduino. Rangkaian modul komunikasi ini dapat dilihat pada Gambar 3.18

3.4 Kontrol PID Proporsional dengan Metode Ziegler Nichols

PID (dari singkatan bahasa Proportional-Integral-Derivative controller) merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tersebut. Komponen kontrol PID ini terdiri dari tiga jenis yaitu Proportional, Integratif dan Derivatif. Ketiganya dapat dipakai bersamaan maupun sendiri-sendiri tergantung dari respon yang diinginkan terhadap suatu plant.

Pada tugas akhir ini akan digunakan kontrol proporsional. Pengaruh kontroler proporsional (P) pada sistem :

1. Menambah atau mengurangi kestabilan.
2. Dapat memperbaiki respon transien khususnya : rise time, settling time
3. Mengurangi (bukan menghilangkan) *Error steady state*.

untuk menghilangkan *Ess*, dibutuhkan *KP* besar, yang akan membuat sistem lebih tidak stabil. Kontroler Proporsional memberi

pengaruh langsung (sebanding) pada *error*. Semakin besar *error*, semakin besar sinyal kendali yang dihasilkan kontroler.

3.4.1 Aturan Ziegler-Nichols

Terkadang pemodelan matematis suatu plant susah untuk dilakukan. Jika hal ini terjadi maka perancangan kontroler PID secara analitis tidak mungkin dilakukan sehingga perancangan kontroler PID harus dilakukan secara eksperimental. Pada bagian ini akan dibahas tentang perancangan kontroler PID secara eksperimental dengan menggunakan aturan Ziegler – Nichols.

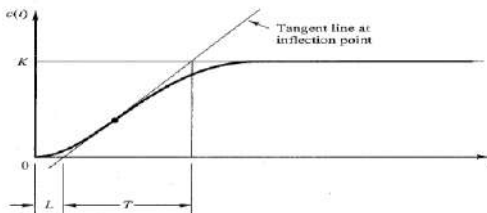
Ziegler dan Nichols memberikan aturan untuk menentukan nilai penguatan proporsional K_p , waktu integral τ_i , dan waktu differensial τ_d yang didasarkan pada karakteristik respon transien dari plant. Terdapat 2 metode penentuan nilai parameter dari ziegler Nichols.

1. Metode Pertama Ziegler Nichols

Dalam metode pertama, kita perlu mendapatkan respon plat terhadap masukan sinyal step. Jika plant tidak mengandung integrator atau kutub pasangan kompleks yang dominan, maka kurva respon step plant tersebut kelihatan seperti kurva bentuk S..

Jika respon plant tidak memiliki kurva berbentuk S, metode ini tidak berlaku. Kurva respon step dapat dihasilkan secara eksperimen atau dari simulasi dinamik sistem.

Kurva S-bentuk dapat dicirikan oleh dua konstanta, waktu tunda L dan konstanta waktu T . Waktu tunda dan konstanta waktu ditentukan dengan menggambar garis singgung pada titik infleksi kurva berbentuk S dan menentukan persimpangan tangen sejajar dengan sumbu waktu dan garis $c(t) = K$, seperti yang ditunjukkan pada Gambar 3.19.



Gambar 3.19 Step Respon Metode Ziegler Nichols

Tabel 3.1 Aturan Ziegler Nichols Berdasarkan Step Respon dari Plant

Type of Controller	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	$2L$	$0.5L$

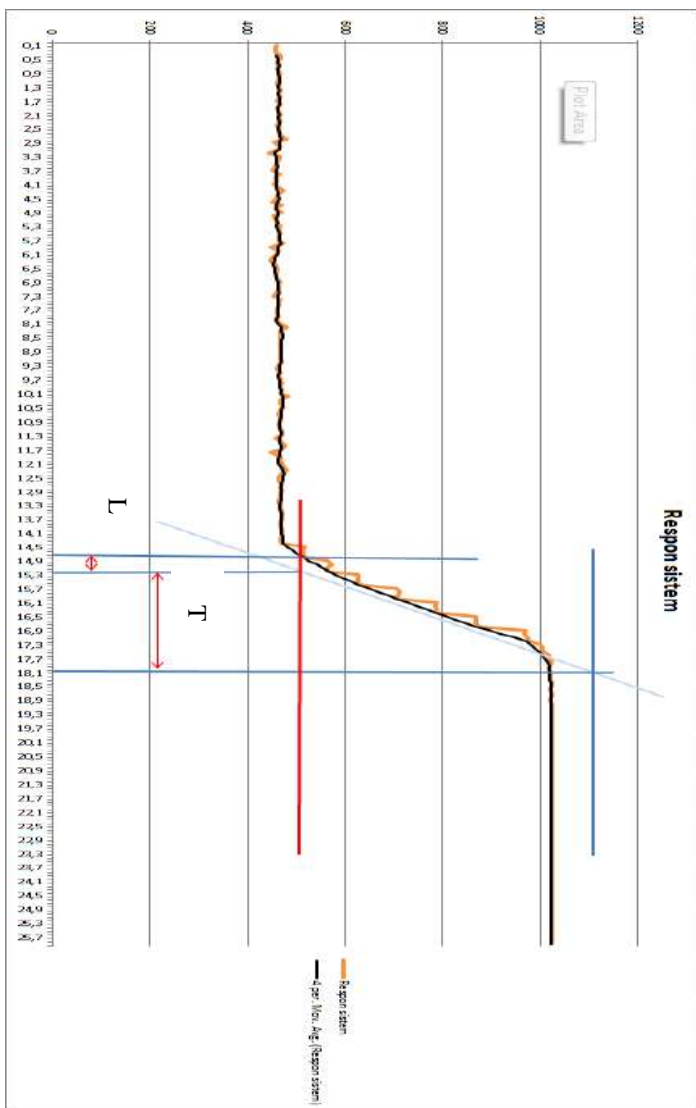
fungsi $C(s) / U(s)$ kemudian dapat didekati oleh sistem orde pertama dengan jeda transportasi sebagai berikut:

$$\frac{C_S}{U_S} = \frac{Ke^{-Ls}}{T_S + 1}$$

Ziegler dan Nichols menyarankan untuk menetapkan nilai-nilai K_p , τ_i dan τ_d sesuai dengan rumus yang ditunjukkan pada tabel 3.2. Perhatikan bahwa kontroler PID disetel oleh metode pertama dari aturan Ziegler-Nichols.

$$\begin{aligned} G_C(s) &= K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \\ &= 1.2 \frac{T}{L} \left(1 + \frac{1}{2L_s} + 0.5L_s \right) \\ &= 0.6T \frac{\left(s + \frac{1}{L} \right)^2}{s} \end{aligned}$$

Dengan demikian, kontroler PID memiliki kutub pada titik asal dan nol ganda pada $s = -1/L$,



Gambar 3.20 Grafik Respon Sistem

Tabel 3.2 Nilai Parameter Ziegler-Nichols

	kp	ki	kd
p	5,6	0	0
pi	5,04	1,666667	0
pid	6,72	1	0,25

Pada Gambar 3.20 didapatkan grafik respon dari motor servo. Dari grafik tersebut dapat diambil nilai L dan T untuk penentuan parameter kontrol proporsional. Dari grafik tersebut didapatkan nilai L=0,5 dan T=2,5, sehingga jika dimasukkan pada tabel Ziegler-Nichols didapatkan nilai seperti pada tabel 3.2.

3.5 Perancangan Perangkat Lunak

Dalam perangkat lunak, terdapat beberapa program yang harus dibuat agar dapat terbaca nilai dari pembacaan sensor yang berada pada sisi operator maupun pada sisi robot.

3.5.1 Perangkat Lunak Sisi Operator

Pada sub bab ini dibahas *flowchart* program pada sisi master beserta penjelasan program yang digunakan. *Flowchart* dapat dilihat pada Gambar 3.21. Pada *flowchart* tersebut dijelaskan bagaimana awal dari program bekerja sampai bagaimana system mendapatkan data dari sensor – sensor. Diawali dengan inisialisasi setiap variable sensor yaitu berupa sensor IMU dan juga sensor *flex*. Setelah diinisialisasi kemudian diberi nilai awal atau acuan dari masing - masing sensor. Setelah itu system akan mengolah data yang didapatkan dari sensor. Misalnya pada data dari MPU6050 akan diolah dengan cara diberikan *low pass filter* kemudian diubah menjadi sudut. Sedangkan untuk sensor *flex* yaitu dengan cara memetakan data adc dari sensor menuju sudut 0° sampai 180°.

Setelah proses pengakuisisian data dan pemrosesan data selesai maka langkah berikutnya yaitu mengirimkan data tersebut menggunakan modul *wireless* Bluetooth HC-05. Setelah data terkirim proses selanjutnya akan dikerjakan pada bagian sisi lengan robot.



Gambar 3.21 Flowchart Program pada Sisi Operator

```

#include<Wire.h>
int sensorvalue =0;
float yaw,yawsold,rollsold,pitchsold,sudutsold;
float yaws,rolls,pitchs,suduts;
const int MPU_addr=0x68
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
unsigned long int data, now;
int mapflex;
  
```

Pada program diatas dijelaskan beberapa penginisialisasian sensor flex, MPU6050. Untuk mendapatkan raw data dari MPU digunakan variabel yaw, yawssold, rollssold, pitchold, sudutsold, yaws, rolls, pitches, dan suduts. Variabel berikut berupa float yang berarti data bilangan desimal. Variabel const int MPU_addr diatas menjelaskan bahwasannya *device addres* dari MPU6050 yaitu berada pada 0x68. Kemudian int16_t tersebut menjelaskan variabel accelerometer dan gyrometer yang akan diakuisisi. Untuk mapflex yaitu variabel untuk menampung data dari pembacaan sensor flex.

```
void setup(){
Wire.begin();
Wire.beginTransmission(MPU_addr);
Wire.write(0x6B); // PWR_MGMT_1 register
Wire.write(0); // set to zero (wakes up the MPU-6050)
Wire.endTransmission(true);
Wire.begin();
Wire.beginTransmission(MPU_addr);
Wire.write(0x1b); //gyro config
Wire.write(0b00001000); //fs_sel 500 degree/s is 32768 (16bit)
Wire.endTransmission();
pinMode(A0,INPUT);
pinMode(3,OUTPUT);
Serial.begin(38400);
}
```

Pada void setup diatas menjelaskan bahwasannya program hanya akan di eksekusi sekali saja. Wire begin yaitu perintah untuk memulai I2C. Kemudian wire begintransmission yaitu perintah untuk mengakses MPU address. Kemudian Wire.write(0x6B) digunakan untuk memulai power management dari MPU6050. Kemudian perintah Wire.write(0); set to zero (wakes up the MPU-6050) dan Wire.endTransmission(true); yaitu memulai mengambil data pembacaan data accelerometer.

Dengan perintah yang sama diulang kembali untuk mendapatkan data dari gyrometer. Kemudian perintah pinMode digunakan untuk menginisialisasi pin sebagai *input* atau *output*.

```

void loop(){
Wire.beginTransmission(MPU_addr);
Wire.write(0x3B);      // starting with register 0x3B
(ACCEL_XOUT_H)
Wire.endTransmission(false);
Wire.requestFrom(MPU_addr,14,true); // request a total of 14
registers
AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) &
0x3C (ACCEL_XOUT_L) 59
AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H)
& 0x3E (ACCEL_YOUT_L) 61
AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) &
0x40 (ACCEL_ZOUT_L) 63
Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) &
0x42 (TEMP_OUT_L)
GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) &
0x44 (GYRO_XOUT_L)
GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) &
0x46 (GYRO_YOUT_L)
GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) &
0x48 (GYRO_ZOUT_L)
}

```

Program diatas menjelaskan data apa saja yang akan diambil nilainya. Pada program diatas data accelerometer temperature dan gyrometer adalah data yang diambil.

Pada program dibawah dijelaskan beberapa proses diantaranya pembacaan sensor flex, dilanjutkan program konversi data raw dari mpu6050 menjadi data dalam bentuk sudut. Kemudian data yang sudah dirubah dalam bentuk sudut di filter menggunakan program exponential filter

```

//=====ambil data flex=====
sensorvalue = analogRead(A0);
mapflex= (int) map(sensorvalue,600,900,2500,500);
//float sudut = (sensorvalue-250)/2;

//=====data pitch roll yaw=====
float pitch = atan2 (AcY, AcZ) * 57.2958;
float roll = atan2 (AcX,AcZ) * 57.2958;
float yawDPS = (float) GyZ/65.356; //raw to dps
yaw = yaw + yawDPS/250+0.00404;

//filtering data (smoothing)
yawsold=yaws;
yaws=yawsold + 0.1*(yaw-yawsold);

rollsold=rolls;
rolls=rollsold + 0.1*(roll-rollsold);

pitchesold=pitch;
pitch=pitchesold + 0.1*(pitch-pitchesold);

```

Pada program dibawah dijelaskan proses pengiriman data data dari sensor MPU6050 dan sensor flex. Data dari sensor MPU yang dikirim berupa data PITCH danYAW. Selain data MPU6050 data yang dikirim juga terdapat data sensor flex.

```
Serial.write("X");
int maproll= (int) map(rolls,-170,170,0,2000);
byte MSBR = (byte) ((maproll >> 8)&0xFF);
byte LSBR = (byte) ((maproll)&0xFF);
Serial.write(MSBR);
Serial.write(LSBR);

int mapyaw= (int) map(yaws,-140,110,0,2000);
byte MSBY = (byte) ((mapyaw >> 8)&0xFF);
byte LSBY = (byte) ((mapyaw)&0xFF);
Serial.write(MSBY);
Serial.write(LSBY);

mapflex= constrain(mapflex,0,2000);
byte MSBF = (byte) ((mapflex >> 8)&0xFF);
byte LSBF = (byte) ((mapflex)&0xFF);
Serial.write(MSBF);
Serial.write(LSBF);
```

3.5.2 Perangkat Lunak Sisi Lengan Robot

Pada sub bab ini dibahas *flowchart* program pada sisi Robot beserta penjelasan program yang digunakan. *Flowchart* dapat dilihat pada Gambar 3.20



Gambar 3.22 *Flowchart* Program Sisi Robot


```

#include <Wire.h>
unsigned int roll,yaw,gripper;
int rollPin = 10;
int pulse1 = 1500;
int yawPin = 11;
int pulse2 = 1500;
int gripperPin = 9;
int pulse3 = 1500;
int servo = 6;
int pulse4 = 700;
long previousMillis = 0;
long previousMillis2 = 0;
long interval = 20;
long interval2 = 50;
int sensor_posisi = A0;
int sensor_gas = A1;
int sensor_value1 = 0;
int sensor_value2 = 0;
int conv_sudut;
int ledState = LOW;

```

Pada program diatas menjelaskan inialisasi nilai posisi awal dari servo, yaitu pada pulse1, pulse2, pulse3, pulse4. Kemudian inialisasi pin pwm yang digunakan yaitu pada rollpin, yawpin, gripperpin, dan servo. Kemudian inialisasi pin *input* dari sensor gas yaitu pada pin sensor_gas = A1, kemudian sensor posisi pada pin A0. Kemudian sensor_value1 dan sensor_value2 digunakan sebagai variabel penampung data dari pembacaan ADC sensor.

```

void setup() {
  Serial.begin(38400);
  pinMode(rollPin,OUTPUT);
  pinMode(yawPin,OUTPUT);
  pinMode(gripperPin,OUTPUT);
  pinMode(13,OUTPUT);
}

```

Pada program void setup diatas dijelaskan bahwa program hanya dieksekusi 1 kali saja. Isi program pada void setup yaitu perintah untuk membuat pin 9, 10, dan 11 digunakan sebagai *output*.

```
void loop() {
  if(Serial.available()>8){
    char inchar = Serial.read();
    if (inchar == 'X'){
      byte MSBR = Serial.read();
      byte LSBR = Serial.read();
      roll = word(MSBR,LSBR);

      byte MSBY = Serial.read();
      byte LSBY = Serial.read();
      yaw = word(MSBY,LSBY);

      byte MSBF = Serial.read();
      byte LSBF = Serial.read();
      gripper = word(MSBF,LSBF);

      pulse1 = (int)map(roll,0,2000,500,2500);
      pulse2 = (int)map(yaw,0,2000,500,2500);
      pulse3 = (int)map(gripper,0,2000,500,2500);
    }
  }
}
```

Pada program diatas dijelaskan proses menerima data dari sisi master. Data dari master tersebut kemudian langsung di konversi kedalam pergerakan motor servo berupa perintah pulse1, pulse2 dan pulse3.

```

sensor_value1 = analogRead(sensor_posisi);
float tegangan = (float)(sensor_value1*5.0/1023.0);
float sudut = (tegangan*300.0)/5.0;
int balik_nilai = map (sudut,0,300,300,0);
conv_sudut = map (balik_nilai,0,300,500,2500);

unsigned long currentMillis2 = millis();
sensor_value2 = analogRead(sensor_gas);
if (sensor_value2 > 300){
  if (currentMillis2 - previousMillis2 >= interval2){
    previousMillis2 = currentMillis2;
    if (ledState == LOW) {
      ledState = HIGH;
    }
    else {
      ledState = LOW;
    }
    digitalWrite(13, ledState);
  }
}
else{
  digitalWrite(13, LOW);
}

unsigned long currentMillis = millis();
if (currentMillis - previousMillis > interval){
  previousMillis = currentMillis;
  updateServo(rollPin,pulse1);
  updateServo(yawPin, pulse2);
  updateServo(gripperPin, pulse3);
  updateServo(servo, pulse4);
}

```

Pada program diatas dijelaskan proses pengolahan nilai sudut yang dihasilkan dari flex sensor dan sensor posisi potensiometer. Kemudian program selanjutnya yaitu proses pengolahan data dari sensor mq2 yang apabila nilai dari sensor melebihi threshold maka akan menyalakan buzzer on off dengan selang waktu 50ms.

```
void updateServo(int pin, int pulse){  
    digitalWrite(pin,HIGH);  
    delayMicroseconds(pulse);  
    digitalWrite(pin,LOW);  
}
```

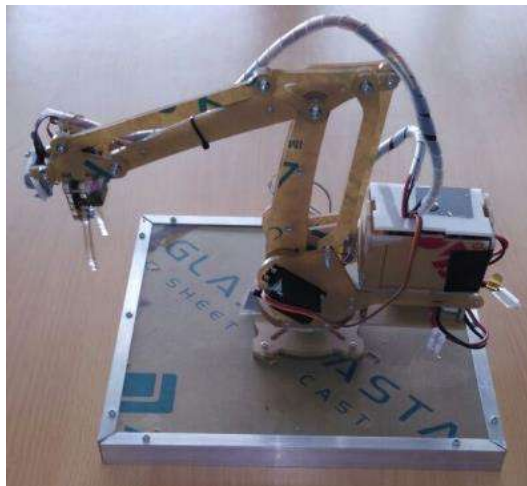
Pada program diatas dijelaskan bahwasannya subfunction program berikut bertugas untuk membuat timer pada servo sehingga servo dapat bergerak dengan baik.

-----Halaman ini sengaja dikosongkan-----

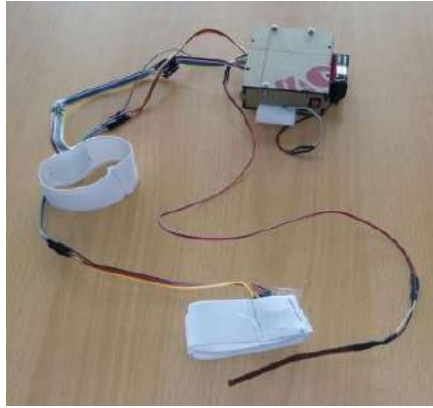
BAB IV PENGUKURAN DAN PENGUJIAN

Untuk mengetahui bahwa alat telah bekerja dengan benar maka perlu dilakukan pengujian alat yang meliputi pengujian perangkat keras dan pengujian perangkat lunak. Pengujian yang dilakukan pada peralatan untuk mengetahui kesesuaian antara teori dengan hasil perancangan, yaitu dengan mengetahui hasil pengukuran pada setiap perangkat yang telah dibuat. Alat yang telah selesai dibuat, dapat dilihat pada Gambar 4.1 dan 4.2, alat tersebut secara umum terdapat 2 bagian , lengan robot dan alat yang akan dipasang pada lengan operator.

Kemampuan dari lengan robot yang telah dibuat yaitu dapat bergerak sebanyak 2dof dan 2 axis. Untuk menggerakkan lengan robot ini digunakan sensor IMU dengan seri MPU6050 sebanyak 2 buah. Salah satu dari sensor IMU tersebut digunakan sebagai *referece* atau titik acuan kemudian sensor IMU yang kedua digunakan untuk mendeteksi sudut yang dibentuk dari kedua sensor.



Gambar 4.1 Realisasi Lengan Robot.



Gambar 4.2 Realisasi Alat Pada Lengan Operator

Actuator dari lengan robot yaitu motor servo dengan seri MG995 sebanyak 4 buah. Jangkauan maksimal dari lengan robot yaitu 60 cm dengan kemampuan mengangkat beban yaitu maksimal xx kg. Selain itu lengan robot ini dilengkapi dengan sensor gas dengan tipe MQ2 yang mana sensitivitas tinggi terhadap gas butane. lengan robot ini dilengkapi dengan modul Bluetooth sebagai komunikasi secara *wireless* dengan jarak maksimal pengiriman data yaitu 5 meter tanpa terhalang benda apapun.

4.1 Pengukuran dan Pengujian Sensor MPU6050

Tujuan dari pengukuran dan pengujian sensor MPU6050 adalah mengetahui kerja dari sensor MPU6050. Cara pengambilan data yang digunakan adalah memprogram Arduino dan membacanya dengan komunikasi serial.

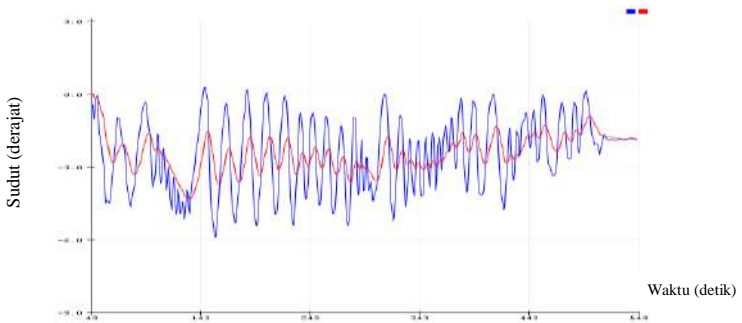
Setelah didapatkan raw data maka pengujian berikutnya bertujuan melihat kesesuaian sudut yang dihasilkan oleh MPU6050 terhadap sudut yang diberikan. Dengan memberikan filter pada *output* yang dikeluarkan oleh MPU6050 maka akan dihasilkan nilai yang lebih baik. Perbandingan nilai antara sebelum dan sesudah di filter ditunjukkan pada Gambar 4.1 dan 4.2.

Pada Gambar 4.1 menunjukkan perbandingan antara nilai *output* MPU6050 sumbu yaw sebelum dan sesudah difilter. Pada Gambar 4.1

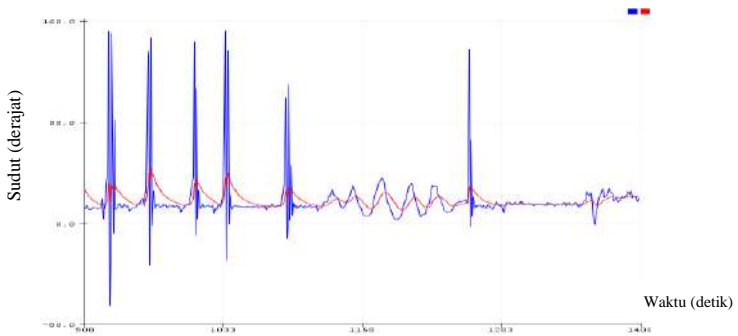
garis biru menunjukkan nilai *output* sebelum di filter dan garis merah menunjukkan nilai *output* sesudah di filter.

Pada Gambar 4.2 menunjukkan perbandingan antara nilai *output* MPU6050 sumbu pitch sebelum dan sesudah difilter. Pada Gambar 4.2 garis biru menunjukkan nilai *output* sebelum di filter dan garis merah menunjukkan nilai *output* sesudah di filter.

$$|Error| = \frac{(nilai\ awal - nilai\ akhir)}{nilai\ awal} \times 100\% \quad (4.1)$$



Gambar 4.3 Data Yaw Sebelum Dan Sesudah Melewati Filter



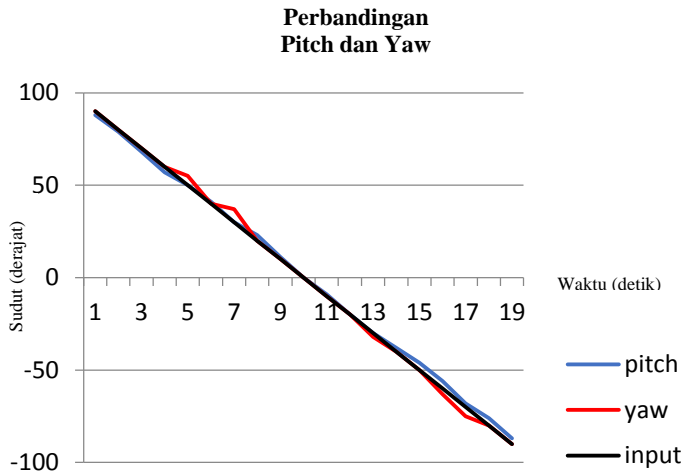
Gambar 4.4 Data Pitch Sebelum Dan Sesudah Melewati Filter

Tabel 4.1 Pengujian Kesesuaian Sudut Pitch dan Yaw

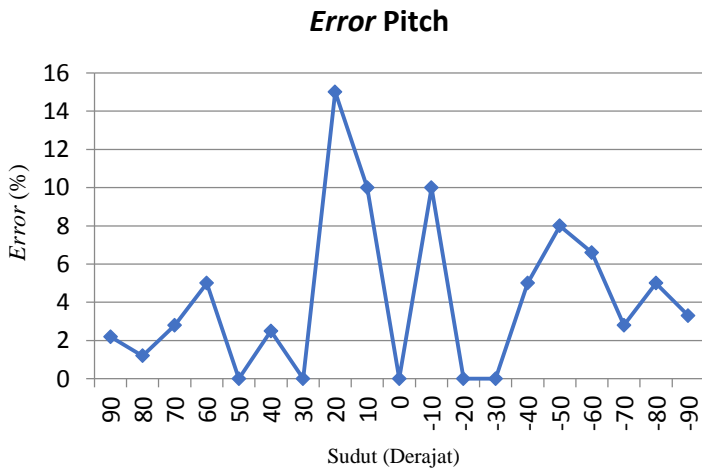
Sudut Input (derajat)	Pitch	Yaw	Error Pitch	Error Yaw
90	88	90	2,2	0
80	79	80	1,2	0
70	68	70	2,8	0
60	57	60	5	0
50	50	55	0	10
40	41	40	2,5	0
30	30	37	0	3,3
20	23	20	15	0
10	11	10	10	0
0	0	0	0	0
-10	-9	-10	10	0
-20	-20	-20	0	0
-30	-30	-32	0	6,6
-40	-38	-40	5	0
-50	-46	-50	8	0
-60	-56	-63	6,6	5
-70	-68	-75	2,8	7,1
-80	-76	-80	5	0
-90	-87	-90	3,3	0
<i>Error Rata- Rata</i>			4,1	0,2

Setelah proses pemfilteran langkah selanjutnya yaitu pengujian kesesuaian gerak dari motor servo. Hasil dari pengujian tersebut dapat dilihat pada tabel 4.1. Ketidaksesuaian pergerakan antara nilai set point dengan realita disebut sebagai *error*. Nilai *error* dapat dihitung dengan persamaan 4.1.

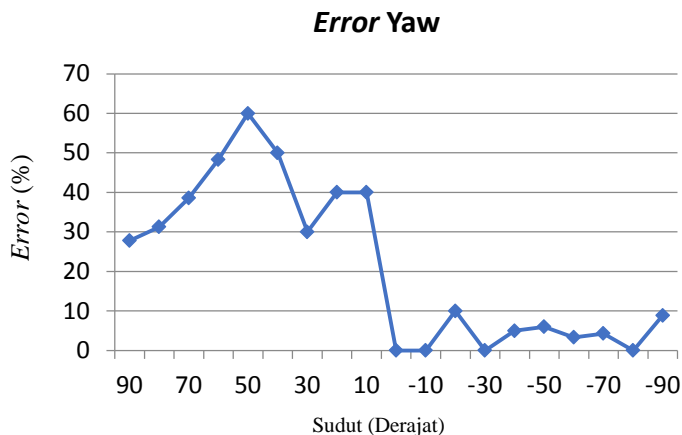
Terlihat pada table 4.1 nilai *error* yang diperoleh pada sumbu pitch dan yaw masing- masing adalah 4,1 dan 0,2 %. Dari hasil tersebut dapat disimpulkan pergeseran dari konversi pergerakan lengan operator menuju pergerakan lengan robot hanya memiliki nilai *error* yang kecil.



Gambar 4.5 Grafik Perbandingan *Input* dan *Output* Pitch, Yaw



Gambar 4.6 Grafik *Error* pada Sumbu Pitch



Gambar 4.7 Grafik *Error* Pada Sumbu Yaw

4.2 Pengukuran dan Pengujian Sensor Flex

Tujuan dari pengukuran dan pengujian sensor flex adalah mengetahui kerja dari sensor flex. Cara pengambilan data yang digunakan adalah memprogram Arduino dan membacanya dengan komunikasi serial.

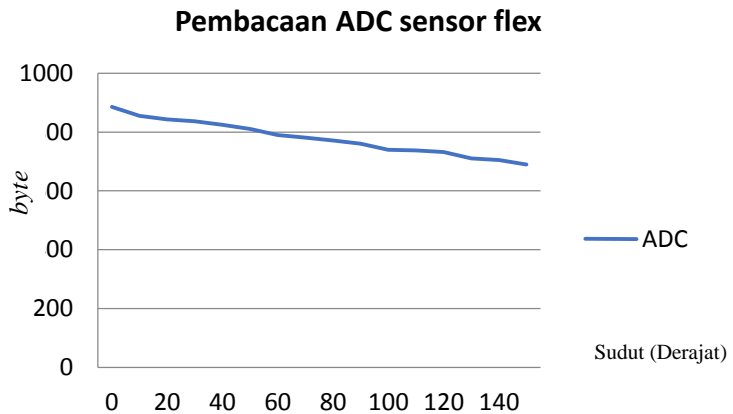
Cara pengujian dari sensor ini yaitu dengan cara menekuk sensor sebesar sudut yang diinginkan kemudian dilakukan pembacaan nilai ADC dari sensor. Setelah didapatkan data pembacaan ADC kemudian dilakukan mapping sudut dari 0 derajat hingga 60 derajat, hal ini dikarenakan pada mekanik penggemang hanya mampu terbuka selebar 60 derajat. Data pengujian dari sensor flex dapat dilihat pada Tabel 4.2.

Dari pengujian sensor flex didapatkan bahwa nilai ADC berbanding terbalik dengan nilai resistansi dari sensor flex, sehingga apabila nilai resistansi dari sensor flex menurun maka pembacaan ADC dari mikrokontroler akan semakin naik.

Dari data pada Tabel 4.2 dapat dilihat bahwa rentang perubahan nilai ADC yaitu 105. Sedangkan perubahan sudut pada servo berkisar 60 derajat. Untuk mendapatkan konversi sudut maka diskalakan perubahan 105 menjadi 0 sampai 60 derajat.

Tabel 4.2 Data Pengujian Sensor Flex







Sudut (derajat)	ADC	Sudut konversi
0	885	0
10	855	10
20	843	15
30	836	30
40	825	0
50	810	0
60	790	0
70	781	0
80	771	0
90	760	0
100	740	10
110	738	10
120	732	20
130	710	20
140	705	30
150	690	30



Gambar 4.1 pembacaan ADC sensor flex

4.3 Pengujian Motor Servo

Tabel 4.3 Hasil pengamatan signal motor servo

Pin	Sinyal PWM	Gambar Osiloskop	Posisi Motor Servo
5	1ms		
9	1.5ms		
10	2ms		

Dalam pengujian kali ini, pengendalian motor servo dilakukan dengan cara mengirimkan sinyal frekuensi sebesar 50 Hz dan dengan waktu TON 1 – 2 ms. Dengan ketentuan bahwa jika pin kontrol motor servo diberikan pulsa 1,5 ms maka motor servo akan bergerak netral. Jika pin kontrol motor servo diberikan pulsa kurang dari 1,5 ms maka motor servo akan bergerak ke arah 0°, dan jika pin kontrol motor servo

diberikan pulsa lebih dari 1,5 ms maka motor servo akan bergerak ke arah 180°.

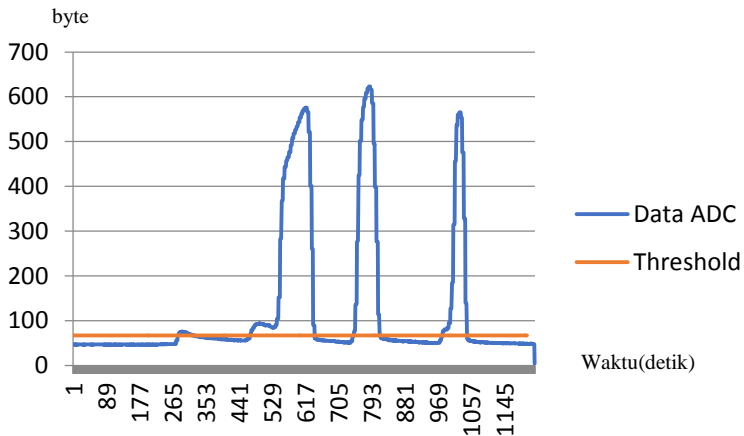
Dari hasil pengamatan dengan osiloskop, pemberian sinyal motor servo pada masing-masing pin, dapat dilihat pada Tabel 4.3.

4.4 Pengukuran dan Pengujian Sensor Gas

Tujuan dari pengukuran dan pengujian dari sensor gas ini adalah mengetahui kerja dari sensor gas MQ2. Cara pengambilan data yang digunakan yaitu dengan menggunakan arduino kemudian pembacaan datanya menggunakan serial monitor arduino.

Gas yang digunakan untuk pengujian yaitu butana. Proses pengambilan datanya adalah dengan menyempatkan gas secara perlahan lalu dilihat data adc yang terbaca pada serial monitor arduino. Berikut data pengambilan dari sensor gas dapat dilihat pada Gambar 4.8.

Pemilihan nilai *threshold* didasarkan pada tingkat konsentrasi kepekatan suatu gas. Nilai *threshold* diambil nilai saat sensor mulai mendeteksi adanya gas. Pada kondisi normal adc terbanya 46 namun ketika mulai diberi gas adc naik pada angka 67. Sehingga penentuan *threshold* yaitu berdasarkan nilai ADC sesaat setelah diberi gas, yaitu pada 67 adc. Sehingga pada pengujian inkator *buzzer* dapat dilihat pada Tabel 4.5.

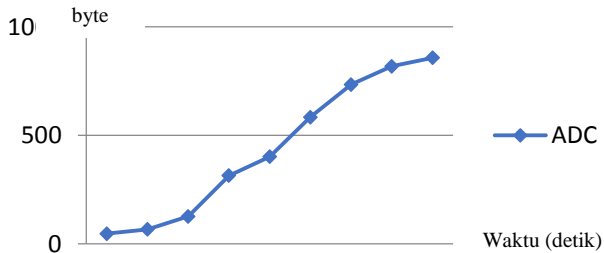


Gambar 4.8 Data Pembacaan Sensor MQ2

Tabel 4.4 Data Pengujian Sensor Gas MQ2

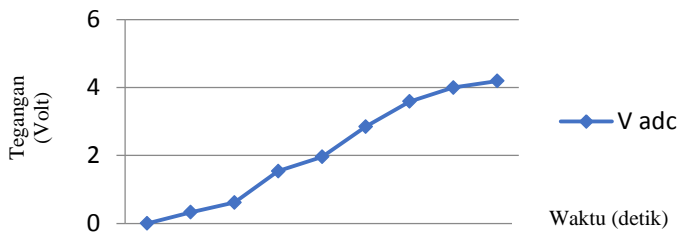
Tipe gas	ADC	Tegangan (V)	Keterangan (buzzer)
Butana	47	0	Mati
	67	0.33	Nyala
	127	0.62	Nyala
	315	1.54	Nyala
	402	1.96	Nyala
	584	2.85	Nyala
	734	3.59	Nyala
	818	4.00	Nyala
	857	4.19	Nyala

Pembacaan ADC sensor MQ2



Gambar 4.9 Grafik Pembacaan Nilai ADC

Pembacaan Tegangan MQ2



Gambar 4.10 Grafik Pembacaan Tegangan ADC

4.5 Pengukuran Dan Pengujian Jarak Pengiriman Data

Tujuan dari pengukuran dan pengujian Jarak adalah mengetahui kemampuan dari modul Bluetooth dalam mengirimkan data. Cara pengambilan data yang digunakan yaitu dengan mengirimkan data untuk menggerakkan lengan robot. Kemudian dari lengan robot dilihat apakah pergerakan sesuai dengan pergerakan lengan operator.

Tabel 4.5 Pengujian Pengiriman Data

Jarak (m)	Keterangan
1	Tersambung
2	Tersambung
3	Tersambung
4	Tersambung
5	Tersambung
6	Terputus
7	Terputus
8	Terputus
9	Terputus
10	Terputus

-----Halaman ini sengaja dikosongkan-----

BAB V

KESIMPULAN DAN SARAN

5.1 KESIMPULAN

Berdasarkan hasil perancangan, pengujian dan hasil analisa dari penelitian yang telah dilakukan dalam Tugas Akhir ini, dapat disimpulkan bahwa, pada Tugas Akhir ini dihasilkan suatu lengan robot dengan derajat kebebasan sebanyak 2 DoF dan 2 Axis pergerakan. Selain itu juga dihasilkan suatu alat kontrol untuk menggerakkan lengan robot. Pada lengan robot dilengkapi pula 1 *gripper* dengan derajat kebebasan yaitu 1 DoF 1 Axis. Jangkauan dari lengan robot yaitu maksimal pada jarak 60 cm. Pada lengan robot ini dilengkapi modul Bluetooth HC-05 yang digunakan sebagai transfer data dari operator menuju lengan robot. Jarak maksimal dari pengiriman data menggunakan modul Bluetooth adalah 5 meter. Proses penentuan sudut dari lengan robot ini menggunakan sensor MPU6050 yang diambil datanya pada sumbu *pitch* dan *yaw*. Dari hasil pengujian sensor tersebut didapatkan bahwa sensor mampu mendeteksi perubahan sudut pada sumbu *pitch* dengan *error* rata-rata sebesar 4,17 % sedangkan *error* rata-rata untuk sumbu *yaw* sebesar 0,2 %. Lengan robot ini juga dilengkapi dengan sensor gas dengan tipe MQ2 yang digu akan untuk mendeteksi ada dan atau tidak adanya suatu gas yang bersada disekitar lokasi lengan robot. Dari hasil pengujian sensor gas didapatkan nilai *threshold* dari sensor yaitu pada 67 nilai ADC, sehingga sistem akan mendeteksi adanya gas apabila pembacaan sensor gas melebihi nilai 67. Indicator dari pendeteksi gas adalah sebuah *buzzer* yang akan berbunyi ketika mendeteksi adanya suatu gas.

5.2 SARAN

Saran untuk penelitian selanjutnya, sebaiknya ditambah fitur berupa penggunaan sensor IMU yang lebih baik supaya didapatkan perubahan atau dapat menambahkan sensor kompas untuk memperbaiki pembacaan sensor MPU6050 pada sumbu *yaw*. Kemudian deteksi sensor gas tidak hanya mendeteksi ada atau tidak adanya gas tetapi mampu mengetahui jenis gas nya.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] Utomo Priyambodoh. 2017., ”rentetan bom bunuh-diri-di indonesia” : Kumparan.
- [2] Feri Fernandes, 2016, “AKBP Dodi Kenang Peristiwa Bom Buku yang Hilangkan Tangan Kirinya”, :detikNews
- [3] Hani Avrilyantama, Muhammad Rivai, Djoko Purwanto, 2015, “Pengembangan Robot Hexapod untuk Melacak Sumber Gas”, JURNAL TEKNIK ITS Vol. 4, No. 1
- [4] Nadia, A., dan Afnan, S. , 2015 "Tangan Robot Peniru GerakTangan Manusia Berbasis Mikrokontroler", Tugas Akhir, Institut Teknologi Sepuluh Nopember, Surabaya.
- [5] Fahrezi Alwi Muhammad, Muhammad Rivai, dan Suwito, 2016, ”Perancangan Sistem Stabilisasi Kamera Tiga Sumbu dengan Metode Kontrol Fuzzy untuk Mobile Surveillance Robot”, JURNAL TEKNIK ITS Vol. 5, No. 2
- [6] Dimas Arief Rahman K., Muhammad Rivai, Rudy Dikairono, 2016, “Sistem Navigasi Pada Balon Udara Menggunakan GPS dan Kontrol Logika Fuzzy”, JURNAL TEKNIK ITS Vol. 5, No. 2.
- [7] Setiawan,Rachmad, 2008, "Teknik Akuisisi Data.Yogyakarta", Graha ilmu.
- [8] Abi Nawang Gustica, Muhammad Rivai, Tasripan, 2014. “Implementasi Sensor Gas pada Kontrol Lengan Robot untuk Mencari Sumber Gas.” JURNAL TEKNIK POMITS Vol. 3, No. 1
- [9] Datasheet MPU-6050 “MPU-6000 and MPU-6050 product specification”, August 2017.
- [10] Arief Saifuddin, Sumardi, Darjat, 2017, “Perancangan Sistem Kendali Pergerakan Arm Manipulator Berbasis Sensor Inertial Measurement Unit (Imu) Dan Sensor Flex”, TRANSIENT, VOL. 6, NO. 3, ISSN: 2302-9927
- [11] Anggraeni Dwi Haryowati, Heri Sutanto, dan Zaenal Arifin, 2010, ”Rancang Bangun Deteksi Alkohol Pada Urine Dengan Sensor TGS 2620 Berbasis Mikrokontroler AT89S51”, Vol. 12, No. 3, ISSN : 1410 – 9662
- [12] HeeBae Song, Doik Kim, Mignon Park, JoonByung Park, 2008, ”Tele-operation between Human and Robot Arm using Wearable Electronic Device”, 17th IFAC World Congress (IFAC'08)

- [13] Uthayakumar, Jayaramaraja B, Meiyappan AN, Naveen Lakshman L, Ramanathan S, 2015, "Gesture controlled robotic surgical arm (GCRSA)", DOI: 10.7439/ijasar
- [14] ____, ITEAD STUDIO datasheet : Bluetooth HC-05, "Bluetooth to serial port module".
- [15] ____, SPECTRA SYMBOL datasheet : Flex sensor Spesial edition length.
- [16] ____, SEMICONDUCTOR SENSOR FOR GAS datasheet : "Semiconductor for combustible gas".
- [17] ____, INVEN SENSE datasheet : "MPU6000 and MPU6050".
- [18] TT ELECTRONIC datasheet : "Model P232/P233 24mm Rotary Potentiometer".
- [19] ____, TOWERPRO datasheet : "MG995 High Speed Metal Gear Dual Ball Bearing Servo".

LAMPIRAN

Berikut adalah lampiran program keseluruhan dari penelitian yang telah dilakukan. Program terbagi menjadi 2 bagian yaitu bagian Operator dan bagian Lengan Robot.

1. Program pada sisi operator.

```
#include<Wire.h>
const int MPU_addr1=0x68, MPU_addr2=0x69;

int16_t AcX_1,AcY_1,AcZ_1,Tmp_1,GyX_1,GyY_1,GyZ_1;
int16_t AcX_2,AcY_2,AcZ_2,Tmp_2,GyX_2,GyY_2,GyZ_2;

float yaw_1,yawsold_1,rollsold_1,pitchsold_1,sudutsold_1;
float yaws_1,rolls_1,pitchs_1,suduts_1;

float yaw_2,yawsold_2,rollsold_2,pitchsold_2,sudutsold_2;
float yaws_2,rolls_2,pitchs_2,suduts_2;

unsigned long int data, now;
int mapflex;
int sensorvalue =0;

void setup(){
  Wire.begin();
  Wire.beginTransmission(MPU_addr1);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Wire.begin();
  Wire.beginTransmission(MPU_addr1);
  Wire.write(0x1b);
  Wire.write(0b00001000);
  Wire.endTransmission();

  Wire.begin();
  Wire.beginTransmission(MPU_addr2);
```

```

Wire.write(0x6B);
Wire.write(0);
Wire.endTransmission(true);
Wire.begin();
Wire.beginTransmission(MPU_addr2);
Wire.write(0x1b);
Wire.write(0b00001000);
Wire.endTransmission();

pinMode(A0,INPUT);
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);

Serial.begin(38400);
}

void loop(){
  digitalWrite(4,HIGH);
  Wire.beginTransmission(MPU_addr1);
  Wire.write(0x3B); // starting with register 0x3B
  (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr1,14,true); // request a total of 14
  registers
  AcX_1=Wire.read()<<8|Wire.read(); // 0x3B
  (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L) 59
  AcY_1=Wire.read()<<8|Wire.read(); // 0x3D
  (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L) 61
  AcZ_1=Wire.read()<<8|Wire.read(); // 0x3F
  (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L) 63
  Tmp_1=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) &
  0x42 (TEMP_OUT_L)
  GyX_1=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H)
  & 0x44 (GYRO_XOUT_L)
  GyY_1=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H)
  & 0x46 (GYRO_YOUT_L)
  GyZ_1=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H)
  & 0x48 (GYRO_ZOUT_L)

```

```

Wire.beginTransmission(MPU_addr2);
Wire.write(0x3B); // starting with register 0x3B
(ACCEL_XOUT_H)
Wire.endTransmission(false);
Wire.requestFrom(MPU_addr2,14,true); // request a total of 14
registers
AcX_2=Wire.read()<<8|Wire.read(); // 0x3B
(ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L) 59
AcY_2=Wire.read()<<8|Wire.read(); // 0x3D
(ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L) 61
AcZ_2=Wire.read()<<8|Wire.read(); // 0x3F
(ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L) 63
Tmp_2=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) &
0x42 (TEMP_OUT_L)
GyX_2=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H)
& 0x44 (GYRO_XOUT_L)
GyY_2=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H)
& 0x46 (GYRO_YOUT_L)
GyZ_2=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H)
& 0x48 (GYRO_ZOUT_L)

```

```

//=====ambil data flex=====
sensorvalue = analogRead(A0);
mapflex= (int) map(sensorvalue,800,900,2500,500);
//Serial.println(sensorvalue);

```

```

//=====data pitch roll yaw MPU1=====
float pitch_1 = atan2 (AcY_1, AcZ_1) * 57.2958;
float roll_1 = atan2 (AcX_1,AcZ_1) * 57.2958;
float yawDPS_1 = (float) GyZ_1/65.356; //raw to dps
yaw_1 = yaw_1 + yawDPS_1/250+0.00404;

```

```

//=====data pitch roll yaw MPU2=====
float pitch_2 = atan2 (AcY_2, AcZ_2) * 57.2958;
float roll_2 = atan2 (AcX_2,AcZ_2) * 57.2958;
float yawDPS_2 = (float) GyZ_2/65.356; //raw to dps

```



```

yaw_2 = yaw_2 + yawDPS_2/250+0.00404;

//filtering data MPU1(smoothing)
yawsold_1=yaws_1;
yaws_1=yawsold_1 + 0.1*(yaw_1-yawsold_1);
rollsold_1=rolls_1;
rolls_1=rollsold_1 + 0.1*(roll_1-rollsold_1);
pitchsold_1=pitchs_1;
pitchs_1=pitchsold_1 + 0.1*(pitch_1-pitchsold_1);
// Serial.println(yaws_1);
// Serial.print("\t");
// Serial.print(pitch_1);
// Serial.print("\t");
// Serial.print(roll_1);
// Serial.print("\t");
// Serial.print("||");
// Serial.print("\t");

//filtering data MPU2(smoothing)
yawsold_2=yaws_2;
yaws_2=yawsold_2 + 0.1*(yaw_2-yawsold_2);
rollsold_2=rolls_2;
rolls_2=rollsold_2 + 0.1*(roll_2-rollsold_2);
pitchsold_2=pitchs_2;
pitchs_2=pitchsold_2 + 0.1*(pitch_2-pitchsold_2);
// Serial.print(yaw_2);
// Serial.print("\t");
// Serial.print(pitch_2);
// Serial.print("\t");
// Serial.print(roll_2);
// Serial.print("\t");
// Serial.print("||");
// Serial.print("\t");

//===Selisih MPU1 - MPU2===
int selisih_yaw= yaws_1 - yaws_2;
int selisih_pitch= pitchs_1 - pitchs_2;
int selisih_roll= rolls_1 - rolls_2;

```

```

// Serial.println(selisih_yaw);
// Serial.print("\t");
// Serial.print(selisih_pitch);
// Serial.print("\t");
// Serial.println(selisih_roll);

//===Mengirim data MPU1===
Serial.write("X");
int maproll_1= (int) map(selisih_roll,-170,170,0,2000);
byte MSBR_1 = (byte) ((maproll_1 >> 8)&0xFF);
byte LSBR_1 = (byte) ((maproll_1)&0xFF);
Serial.write(MSBR_1);
Serial.write(LSBR_1);
// Serial.print(MSBR);
// Serial.println(LSBR);

int mapyaw_1= (int) map(selisih_yaw,-90,50,0,2000);
byte MSBY_1 = (byte) ((mapyaw_1 >> 8)&0xFF);
byte LSBY_1 = (byte) ((mapyaw_1)&0xFF);
Serial.write(MSBY_1);
Serial.write(LSBY_1);
// Serial.print(MSBY);
// Serial.println(mapyaw);

mapflex= constrain(mapflex,0,2000);
byte MSBF = (byte) ((mapflex >> 8)&0xFF);
byte LSBF = (byte) ((mapflex)&0xFF);
Serial.write(MSBF);
Serial.write(LSBF);
// Serial.print(MSBF);
// Serial.println(LSBF);
}

```

2. Program pada sisi lengan robot.

```
#include <Wire.h>
```

```

unsigned int roll,yaw,gripper;
int rollPin = 10;
int pulse1 = 1500;

int yawPin = 11;
int pulse2 = 1500;

int gripperPin = 9;
int pulse3 = 1500;

int servo = 5;
int pulse4 = 750;

long previousMillis = 0;
long previousMillis2 = 0;
long interval = 20;
long interval2 = 50;

int sensor_posisi = A0;
int sensor_gas = A1;
int sensor_value1 = 0;
int sensor_value2 = 0;
int conv_sudut;

int ledState = LOW;

float error, errorI, errorD, error_sblmI,error_sblmD, OutP, OutI,
OutD, Kp, Ki, Kd ;
float Tc=0.01,outPID;

void setup() {
  Serial.begin(38400);
  pinMode (rollPin,OUTPUT);
  pinMode (yawPin,OUTPUT);
  pinMode (gripperPin,OUTPUT);
  pinMode (servo,OUTPUT);
  pinMode (13,OUTPUT);

  Kp= 5.6;

```

```

Ki= 0;
Kd= 0;
error_sblmI= 0;
error_sblmD= 0;
}

void loop() {

if(Serial.available(>8){
char inchar = Serial.read();
if (inchar == 'X'){
byte MSBR = Serial.read();
byte LSBR = Serial.read();
roll = word(MSBR,LSBR);

byte MSBY = Serial.read();
byte LSBY = Serial.read();
yaw = word(MSBY,LSBY);

byte MSBF = Serial.read();
byte LSBF = Serial.read();
gripper = word(MSBF,LSBF);

pulse1 = (int)map(roll,0,2000,500,2500);
pulse2 = (int)map(yaw,0,2000,500,2500);
pulse3 = (int)map(gripper,0,2000,500,2500);

}
}

sensor_value1 = analogRead(sensor_posisi);
float tegangan = (float)(sensor_value1*5.0/1023.0);
float sudut = (tegangan*300.0)/5.0;
int balik_nilai = map (sudut,0,300,300,0);
conv_sudut = map (balik_nilai,0,300,500,2500);
//Serial.println(conv_sudut);

unsigned long currentMillis2 = millis();
sensor_value2 = analogRead(sensor_gas);

```

```

// Serial.println(sensor_value2);
if (sensor_value2 > 67){
  if (currentMillis2 - previousMillis2 >= interval2){
    previousMillis2 = currentMillis2;
    if (ledState == LOW) {
      ledState = HIGH;
    }
    else {
      ledState = LOW;
    }
    digitalWrite(13, ledState);
  }
}
else{
  digitalWrite(13, LOW);
}
}

```

```

unsigned long currentMillis = millis();
if (currentMillis - previousMillis > interval){
  previousMillis = currentMillis;
  updateServo(servo,pulse4);
  updateServo(rollPin,pulse1);
  updateServo(yawPin, pulse2);
  updateServo(gripperPin, pulse3);
}
//Serial.println(outPID);

error = pulse1 - conv_sudut;
pid();
}

```

```

void updateServo(int pin, int pulse){
  digitalWrite(pin,HIGH);
  delayMicroseconds(pulse);
  digitalWrite(pin,LOW);
}

```

```

void pid(){
  OutP = Kp * error;
  if (error == 0)
    error_sbImI = 0;
  errorI = error +error_sbImI;
  OutI = Ki*errorI*Tc;
  error_sbImI = errorI;

  errorD = error-error_sbImD;
  OutD = (Kd*errorD)/Tc;
  error_sbImD = error;

  outPID = OutP + OutI + OutD;
  if (outPID >= 2500)
    outPID = 2500;
  if (outPID <= 500)
    outPID = 500;
  if (error >= 1400 && error <=1600){
    outPID = 1500;
  }
}

```

3. Daftar Komponen Penyusun Gripper

Berikut adalah komponen yang digunakan untuk membuat gripper.

Nama Komponen	Deskripsi komponen	Jumlah
Gripper	Screw M2-10 2	2
	Screw M3-8 4	4
	Screw M3-25 1	1
	Screw M4-16 2	2
	Nut M2 2	2
	Nut M3 2	2
	Lock Nut M4 2	2
	Brass Stand-offs M3-8 2	2
	Metal Spacer 4-10-1	2
	Shaft Sleeve 2.5-3-4 1	1
	Plastic Spacer 3-7-10 1	1

	Plastic Spacer 4-7-5	2
--	----------------------	---

4. Daftar Komponen Penyusun Lengan Robot

Nama Komponen	Deskripsi komponen	Jumlah
Acrylic Parts	Acrylic Board	4
Basic uArm 1	Screw M4-8	16
	Screw M4-16	11
Basic uArm 2	Screw M4-12	12
	Screw M4-22	1
Basic uArm 3	Screw M4-14	4
	Screw M4-25	8
Basic uArm 4	Screw M2-10	2
	Tapping Screw M2-10	8
	Tapping Screw M2-12	4
Basic uArm 5	Nut M2	2
	Nut M4	24
	Lock Nut M4	4
Basic uArm 6	Metal Spacer 4x10x1	5
	Shaft Sleeve M4-7	1
Basic uArm 7	Brass Stand-offs M4-10	4
	Brass Stand-offs M4-15	1
	Brass Stand-offs M4-22	4
	Brass Stand-offs M4-55	2
	Brass Stand-offs M4-65	1
Basic uArm 8	Thin-walled Bearing 61807	1
	Flange Bearing MF84ZZ	12
	Flange Shaft Sleeve	12
Basic uArm 9	Suction Cup 45-M5	4
	Rivet M5-10	4
Basic uArm 10	Screw M4-28	3

5. Data Sheet Sensor Flex



FLEX SENSOR FS

Special Edition Length

Features

- Angle Displacement Measurement
- Bends and Flexes physically with motion device
- Possible Uses
 - Robotics
 - Gaming (Virtual Motion)
 - Medical Devices
 - Computer Peripherals
 - Musical Instruments
 - Physical Therapy
- Simple Construction
- Low Profile

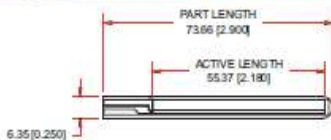
Mechanical Specifications

- Life Cycle: >1 million
- Height: $\pm 0.43\text{mm}$ (0.017")
- Temperature Range: -35°C to +80°C

Electrical Specifications

- Flat Resistance: 25K Ohms
- Resistance Tolerance: $\pm 30\%$
- Bend Resistance Range: 45K to 125K Ohms (depending on bend radius)
- Power Rating : 0.50 Watts continuous, 1 Watt Peak

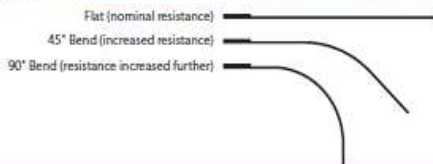
Dimensional Diagram - Stock Flex Sensor



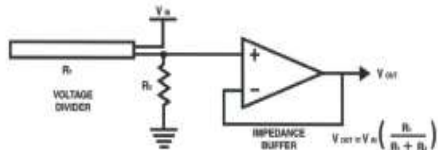
How to Order - Stock Flex Sensor

FS	-	L	-	0055	-	253	-	ST
Series		Model		Active Length		Resistance		Connectors
FS = Flex Sensor		L = Linear		0055 = 55.37mm		253 = 25K Ohms		ST = Solder Tab

How It Works



BASIC FLEX SENSOR CIRCUIT:

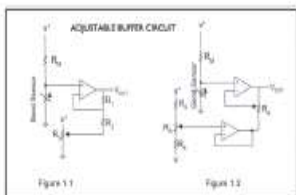


Following are notes from the ITP Flex Sensor Workshop

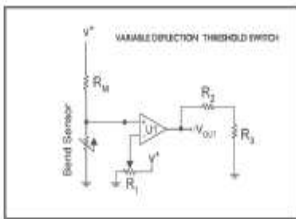
"The impedance buffer in the [Basic Flex Sensor Circuit] (above) is a single sided operational amplifier, used with these sensors because the low bias current of the op amp reduces error due to source impedance of the flex sensor as voltage divider. Suggested op amps are the LM358 or LM324."

"You can also test your flex sensor using the simplest circuit, and skip the op amp."

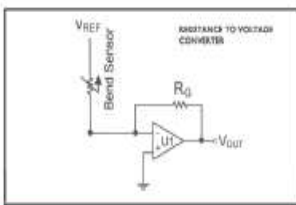
"Adjustable Buffer - a potentiometer can be added to the circuit to adjust the sensitivity range."



"Variable Deflection Threshold Switch - an op amp is used and outputs either high or low depending on the voltage of the inverting input. In this way you can use the flex sensor as a switch without going through a microcontroller."



"Resistance to Voltage Converter - use the sensor as the input of a resistance to voltage converter using a dual sided supply op-amp. A negative reference voltage will give a positive output. Should be used in situations when you want output at a low degree of bending."



6. Data Sheet Sensor MQ2

MQ-2 Semiconductor Sensor for Combustible Gas

Sensitive material of MQ-2 gas sensor is SnO_2 , which with lower conductivity in clean air. When the target combustible gas exist, The sensor's conductivity is more higher along with the gas concentration rising. Please use simple electrocircuit, Convert change of conductivity to correspond output signal of gas concentration.

MQ-2 gas sensor has high sensivity to LPG, Propane and Hydrogen, also could be used to Methane and other combustible steam, it is with low cost and suitable for different application.

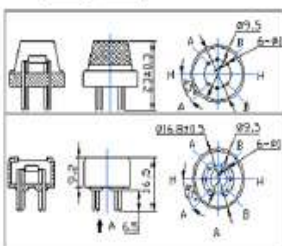
Character

- * Good sensitivity to Combustible gas in wide range
- * High sensitivity to LPG, Propane and Hydrogen
- * Long life and low cost
- * Simple drive circuit

Application

- * Domestic gas leakage detector
- * Industrial Combustible gas detector
- * Portable gas detector

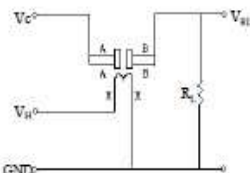
Configuration



Technical Data

Model No.		MQ-2	
Sensor Type		Semiconductor	
Standard Encapsulation		Bakelite (Black Bakelite)	
Detection Gas		Combustible gas and smoke	
Concentration		300-10000ppm (Combustible gas)	
Circuit	Loop Voltage V_L	$\leq 24V$ DC	
	Heater Voltage V_H	5.0V \pm 0.2V AC or DC	
	Load Resistance R_L	Adjustable	
Character	Heater Resistance R_H	31 Ω \pm 3 Ω (Room Tem.)	
	Heater consumption P_H	$\leq 300mW$	
	Sensing Resistance R_A	2K Ω -20K Ω (In 2000ppm C ₂ H ₆)	
	Sensitivity S	$R_S(\text{in air})/R_A(1000ppm \text{ ISOBU2ANE}) \geq 5$	
	Slope α	$\leq 0.5 (R_{1000ppm}/R_{1000ppm} \text{ CH}_4)$	
Condition	Tem. Humidity	20 $^{\circ}C \pm 2^{\circ}C$, 65% \pm 5%RH	
	Standard test circuit	$V_C: 5.0V \pm 0.1V$ $V_H: 5.0V \pm 0.1V$	
	Preheat time	Over 48 hours	

Basic test loop



The above is basic test circuit of the sensor. The sensor need to be put 2 voltage, heater voltage(V_H) and test voltage(V_C). V_H used to supply certified working temperature to the sensor, while V_C used to detect voltage (V_R) on load resistance (R_L) whom is in series with sensor. The sensor has light polarity, V_C need DC power. V_C and V_H could use same power circuit with precondition to assure performance of sensor. In order to make the sensor with better performance, suitable R_L value is needed:
Power of Sensitivity body(P_S):
 $P_S = V_C^2 \times R_L / (R_S + R_L)^2$

Resistance of sensor(R_s): $R_s=(V_c/V_{RL}-1) \times R_{L1}$

Sensitivity Characteristics

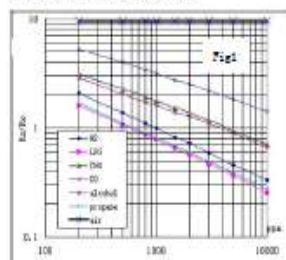


Fig.1 shows the typical sensitivity characteristics of the MQ-2, ordinate means resistance ratio of the sensor (R_s/R_0), abscissa is concentration of gases. R_0 means resistance in different gases, R_0 means resistance of sensor in 1000ppm Hydrogen. All test are under standard test conditions.

Influence of Temperature/Humidity

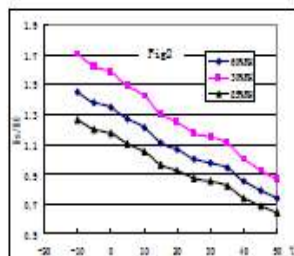
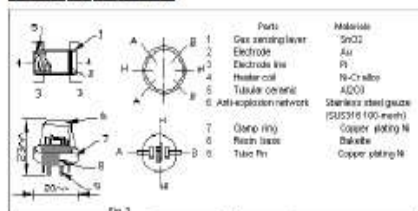


Fig.2 shows the typical temperature and humidity characteristics. Ordinate means resistance ratio of the sensor (R_s/R_0), R_0 means resistance of sensor in 1000ppm Butane under different tem. and humidity. R_0 means resistance of the sensor in environment of 1000ppm Methane, 20°C/65%RH

Structure and configuration



Structure and configuration of MQ-2 gas sensor is shown as Fig. 3, sensor composed by micro Al₂O₃ ceramic tube, Tin Dioxide (SnO₂) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-2 have 6 pin, 4 of them are used to fetch signals, and other 2 are used for providing heating current.

Notification

1 Following conditions must be prohibited

1.1 Exposed to organic silicon steam

Organic silicon steam cause sensors invalid, sensors must be avoid exposing to silicon bond, fixture, silicon latex, putty or plastic contain silicon environment

1.2 High Corrosive gas

If the sensors exposed to high concentration corrosive gas (such as H_2S , SO_x , Cl_2 , HCl etc), it will not only result in corrosion of sensors structure, also it cause sincere sensitivity attenuation.

1.3 Alkali, Alkali metals salt, halogen pollution

The sensors performance will be changed badly if sensors be sprayed polluted by alkali metals salt especially brine, or be exposed to halogen such as fluorin.

1.4 Touch water

Sensitivity of the sensors will be reduced when spattered or dipped in water.

1.5 Freezing

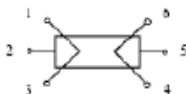
Do avoid icing on sensor's surface, otherwise sensor would lose sensitivity.

1.6 Applied voltage higher

Applied voltage on sensor should not be higher than stipulated value, otherwise it cause down-line or heater damaged, and bring on sensors' sensitivity characteristic changed badly.

1.7 Voltage on wrong pins

For 6 pins sensor, if apply voltage on 1, 3 pins or 4, 6 pins, it will make lead broken, and without signal when apply on 2, 4 pins



2 Following conditions must be avoided

2.1 Water Condensation

Indoor conditions, slight water condensation will effect sensors performance lightly. However, if water condensation on sensors surface and keep a certain period, sensor's sensitivity will be decreased.

2.2 Used in high gas concentration

No matter the sensor is electrified or not, if long time placed in high gas concentration, it will affect sensors characteristic.

2.3 Long time storage

The sensors resistance produce reversible drift if it's stored for long time without electrify, this drift is related with storage conditions. Sensors should be stored in airtight without silicon gel bag with clean air. For the sensors with long time storage but no electrify, they need long aging time for stability before using.

2.4 Long time exposed to adverse environment

No matter the sensors electrified or not, if exposed to adverse environment for long time, such as high humidity, high temperature, or high pollution etc, it will effect the sensors performance badly.

2.5 Vibration

Continual vibration will result in sensors down-lead response then repute. In transportation or assembling line, pneumatic screwdriver/ultrasonic welding machine can lead this vibration.

2.6 Concussion

If sensors meet strong concussion, it may lead its lead wire disconnected.

2.7 Usage

For sensor, handmade welding is optimal way. If use wave crest welding should meet the following conditions:

2.7.1 Soldering flux: Rosin soldering flux contains least chlorine

2.7.2 Speed: 1-2 Meter/ Minute

2.7.3 Warm-up temperature: $100 \pm 20^\circ C$

2.7.4 Welding temperature: $250 \pm 10^\circ C$

2.7.5 1 time pass wave crest welding machine

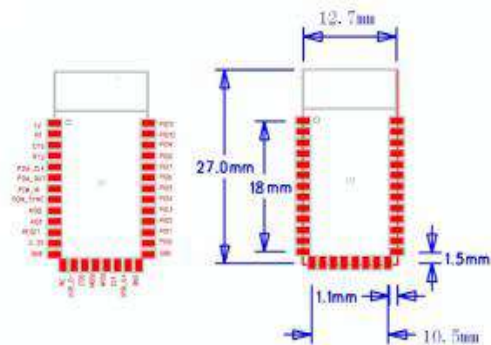
If disobey the above using terms, sensors sensitivity will be reduced.

7. Data Sheet Bluetooth

Software features

- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has.
Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in P100, device will be disconnected.
- Status instruction port P101: low-disconnected, high-connected;
- P1010 and P1011 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- Auto-reconnect in 30 min. when disconnected as a result of beyond the range of connection.

Hardware



PIN Name	PIN #	Pad type	Description	Note
GND	13	VSS	Ground pad	
	21			
	22			
3.3 VCC	12	3.3V	Integrated 3.3V (+) supply with On-chip linear regulator output within 3.15-3.3V	
AIO0	9	Bi-Directional	Programmable input/output line	
AIO1	10	Bi-Directional	Programmable input/output line	
PIO0	23	Bi-Directional RX EN	Programmable input/output line, control output for LNA (if fitted)	
PIO1	24	Bi-Directional TX EN	Programmable input/output line, control output for PA (if fitted)	
PIO2	25	Bi-Directional	Programmable input/output line	
PIO3	26	Bi-Directional	Programmable input/output line	
PIO4	27	Bi-Directional	Programmable input/output line	
PIO5	28	Bi-Directional	Programmable input/output line	
PIO6	29	Bi-Directional	Programmable input/output line	
PIO7	30	Bi-Directional	Programmable input/output line	
PIO8	31	Bi-Directional	Programmable input/output line	
PIO9	32	Bi-Directional	Programmable input/output line	
PIO10	33	Bi-Directional	Programmable input/output line	
PIO11	34	Bi-Directional	Programmable input/output line	

RESETB	11	CMOS input with weak internal pull-up	Reset if low, input debounced so must be low for >5MS to cause a reset.	
UART_RTS	4	CMOS output, tri-stable with weak internal pull-up	UART request to send, active low	
UART_CTS	3	CMOS input with weak internal pull-down	UART clear to send, active low	
UART_RX	2	CMOS input with weak internal pull-down	UART Data input	
UART_TX	1	CMOS output, Tri-stable with weak internal pull-up	UART Data output	
SPI_MOSI	17	CMOS input with weak internal pull-down	Serial peripheral interface data input	
SPI_CSB	16	CMOS input with weak internal pull-up	Chip select for serial peripheral interface, active low	
SPI_CLK	19	CMOS input with weak internal pull-down	Serial peripheral interface clock	
SPI_MISO	18	CMOS input with weak internal pull-down	Serial peripheral interface data Output	
USB_	15	I ² C-Directional		

USB_+	20	Bi-Directional		
NC	14			
PCM_CLK	5	Bi-Directional	Synchronous PCM data clock	
PCM_OUT	6	CMOS output	Synchronous PCM data output	
PCM_IN	7	CMOS Input	Synchronous PCM data input	
PCM_SYNC	8	Bi-Directional	Synchronous PCM data strobe	

AT command Default:

How to set the mode to server (master):

1. Connect PIO11 to high level.
2. Power on, module into command state.
3. Using baud rate 38400, sent the "AT+ROLE=1\r\n" to module, with "OK\r\n" means setting successes.
4. Connect the PIO11 to low level, repower the module, the module work as server (master).

AT commands: (all end with \r\n)

1. Test command:

Command	Respond	Parameter
AT	OK	-

2. Reset

Command	Respond	Parameter
AT+RESET	OK	-


3. Get firmware version

Command	Respond	Parameter
AT+VERSION?	+VERSION:<Param> OK	Param : firmware version

Example:

```
AT+VERSION*\r\n
+VERSION:2.0-20100601
OK
```


8. Data Sheet Sensor MPU6050

	MPU-6000/MPU-6050 Product Specification	Document Number: PS-MPU-6000A-00 Revision: 3.4 Release Date: 08/19/2013
---	--	---

6 Electrical Characteristics

6.1 Gyroscope Specifications

VDD = 2.375V-3.45V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
GYROSCOPE SENSITIVITY						
Full-Scale Range	FS_SEL=0 FS_SEL=1 FS_SEL=2 FS_SEL=3		±250 ±500 ±1000 ±2000		% % % %	
Gyroscope ADC Word Length			16		bits	
Sensitivity Scale Factor	FS_SEL=0 FS_SEL=1 FS_SEL=2 FS_SEL=3		131 65.5 32.8 16.4		LSB/(%g) LSB/(%g) LSB/(%g) LSB/(%g)	
Sensitivity Scale Factor Tolerance	25°C	-3		+3	%	
Sensitivity Scale Factor Variation Over Temperature			±2		%	
Nonlinearity	Best fit straight line, 25°C		0.2		%	
Cross-Axis Sensitivity			±2		%	
GYROSCOPE ZERO-RATE OUTPUT (ZRO)						
Initial ZRO Tolerance	25°C		±20		%	
ZRO Variation Over Temperature	-40°C to +85°C		±20		%	
Power-Supply Sensitivity (1-10Hz)	Sine wave, 100mVpp, VDD=2.5V		0.2		%	
Power-Supply Sensitivity (10 - 250Hz)	Sine wave, 100mVpp, VDD=2.5V		0.2		%	
Power-Supply Sensitivity (250Hz - 100kHz)	Sine wave, 100mVpp, VDD=2.5V		4		%	
Linear Acceleration Sensitivity	Static		0.1		%/g	
SELF-TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	1
GYROSCOPE NOISE PERFORMANCE						
Total RMS Noise	FS_SEL=0 DLPFCFG=2 (100Hz)		0.05		%-rms	
Low-frequency RMS noise	Bandwidth: 1Hz to 10Hz		0.033		%-rms	
Rate Noise Spectral Density	At 10Hz		0.005		%/√Hz	
GYROSCOPE MECHANICAL FREQUENCIES						
X-Axis		30	33	36	kHz	
Y-Axis		27	30	33	kHz	
Z-Axis		24	27	30	kHz	
LOW PASS FILTER RESPONSE						
Programmable Range		5		256	Hz	
OUTPUT DATA RATE						
Programmable		4		8,000	Hz	
GYROSCOPE START-UP TIME						
ZRO Settling (from power-on)	DLPFCFG=0 to ±1% of Final		30		ms	

1. Please refer to the following document for further information on Self-Test: MPU-6000/MPU-6050 Register Map and Descriptions

**6.2 Accelerometer Specifications**VDD = 2.375V-3.45V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
ACCELEROMETER SENSITIVITY						
Full-Scale Range	AFS_SEL=0 AFS_SEL=1 AFS_SEL=2 AFS_SEL=3		±2 ±4 ±8 ±16		g g g g	
ADC Word Length	Output in two's complement format		16		bits	
Sensitivity Scale Factor	AFS_SEL=0 AFS_SEL=1 AFS_SEL=2 AFS_SEL=3		16,384 8,192 4,096 2,048		LSB/g LSB/g LSB/g LSB/g	
Initial Calibration Tolerance			±3		%	
Sensitivity Change vs. Temperature	AFS_SEL=0, -40°C to +85°C		±0.02		%/°C	
Nonlinearity	Best Fit Straight Line		0.5		%	
Cross-Axis Sensitivity			±2		%	
ZERO-G OUTPUT						
Initial Calibration Tolerance	X and Y axes Z axis		±50 ±80		mg mg	1
Zero-G Level Change vs. Temperature	X and Y axes, 0°C to +70°C Z axis, 0°C to +70°C		±35 ±50		mg mg	
SELF TEST RESPONSE						
Relative	Change from factory trim	-14		14	%	2
NOISE PERFORMANCE						
Power Spectral Density	@10Hz, AFS_SEL=0 & ODR=1kHz		400		μg/√Hz	
LOW PASS FILTER RESPONSE						
Programmable Range		5		260	Hz	
OUTPUT DATA RATE						
Programmable Range		4		1,000	Hz	
INTELLIGENCE FUNCTION INCREMENT						
			32		mg/LSB	

1. Typical zero-g initial calibration tolerance value after MSL3 preconditioning
2. Please refer to the following document for further information on Self-Test: MPU-6000/MPU-6050 Register Map and Descriptions

**6.3 Electrical and Other Common Specifications**VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_a = 25°C

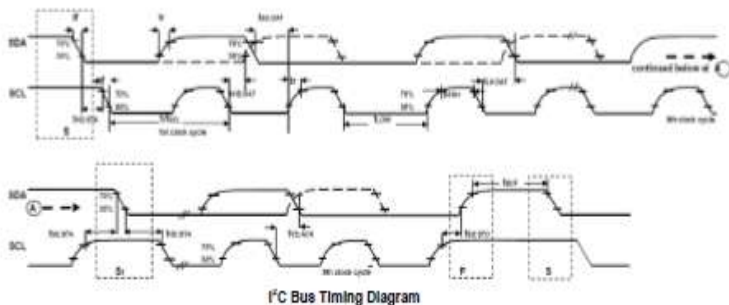
PARAMETER	CONDITIONS	MIN	TYP	MAX	Units	Notes
TEMPERATURE SENSOR						
Range			-40 to +85		°C	
Sensitivity	Untrimmed		±40		LSB/°C	
Temperature Offset	35°C		-521		LSB	
Linearity	Best fit straight line (-40°C to +85°C)		±1		°C	
VDD POWER SUPPLY						
Operating Voltages		2.375		3.46	V	
Normal Operating Current	Gyroscope + Accelerometer + DMP		3.9		mA	
	Gyroscope + Accelerometer (DMP disabled)		3.8		mA	
	Gyroscope + DMP (Accelerometer disabled)		3.7		mA	
	Gyroscope only (DMP & Accelerometer disabled)		3.6		mA	
	Accelerometer only (DMP & Gyroscope disabled)		500		µA	
Accelerometer Low Power Mode Current	1.25 Hz update rate		10		µA	
	5 Hz update rate		20		µA	
	20 Hz update rate		70		µA	
	40 Hz update rate		140		µA	
Full-Chip Idle Mode Supply Current			5		µA	
Power Supply Ramp Rate	Monotonic ramp. Ramp rate is 10% to 90% of the final value			100	ms	
VLOGIC REFERENCE VOLTAGE	MPU-6050 only					
Voltage Range	VLOGIC must be <VDD at all times	1.71		VDD	V	
Power Supply Ramp Rate	Monotonic ramp. Ramp rate is 10% to 90% of the final value			3	ms	
Normal Operating Current			100		µA	
TEMPERATURE RANGE						
Specified Temperature Range	Performance parameters are not applicable beyond Specified Temperature Range	-40		+85	°C	

6.7 I²C Timing Characterization

Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.46V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C

Parameters	Conditions	Min	Typical	Max	Units	Notes
I²C TIMING						
I ² C FAST-MODE						
f _{SCL} , SCL Clock Frequency				400	kHz	
t _{SC,ms} , (Repeated) START Condition Hold Time		0.6			µs	
t _{LOW}} , SCL Low Period		1.3			µs	
t _{HIGH}} , SCL High Period		0.6			µs	
t _{SETUP,ms} , Repeated START Condition Setup Time		0.6			µs	
t _{HD,µs}} , SDA Data Hold Time		0			µs	
t _{DS,ns}} , SDA Data Setup Time		100			ns	
t _{r}} , SDA and SCL Rise Time	C _L bus cap. from 10 to 400pF	20+0.1C _L		300	ns	
t _{f}} , SDA and SCL Fall Time	C _L bus cap. from 10 to 400pF	20+0.1C _L		300	ns	
t _{STOP,ns}} , STOP Condition Setup Time		0.6			µs	
t _{FREE}} , Bus Free Time Between STOP and START Condition		1.3			µs	
C _L , Capacitive Load for each Bus Line			< 400		µF	
t _{DATA}} , Data Valid Time				0.0	µs	
t _{ACK}} , Data Valid Acknowledge Time				0.0	µs	

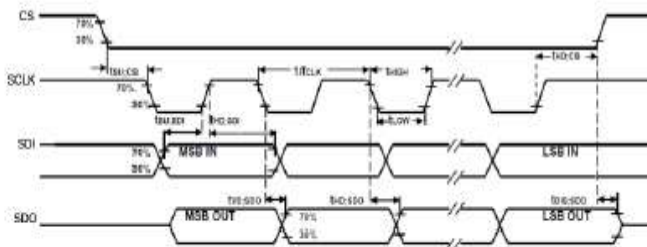
Note: Timing Characteristics apply to both Primary and Auxiliary I²C Bus



6.8 SPI Timing Characterization (MPU-6000 only)

 Typical Operating Circuit of Section 7.2, VDD = 2.375V-3.45V, VLOGIC (MPU-6050 only) = 1.8V±5% or VDD, T_A = 25°C, unless otherwise noted.

Parameters	Conditions	Min	Typical	Max	Units	Notes
SPI TIMING						
f _{CLK} , SCLK Clock Frequency				1	MHz	
t _{LOW} , SCLK Low Period		400			ns	
t _{HIGH} , SCLK High Period		400			ns	
t _{setup,CS} , CS Setup Time		8			ns	
t _{hold,CS} , CS Hold Time		500			ns	
t _{setup,SDI} , SDI Setup Time		11			ns	
t _{hold,SDI} , SDI Hold Time		7			ns	
t _{valid,SDO} , SDO Valid Time	C _{load} = 20pF			100	ns	
t _{hold,SDO} , SDO Hold Time	C _{load} = 20pF	4			ns	
t _{disable,SDO} , SDO Output Disable Time				10	ns	


SPI Bus Timing Diagram

BIODATA PENULIS



Wahyu Satrio Prayogo lahir di Ponorogo, Jawa Timur pada tanggal 18 November 1994. Penulis adalah anak ketiga dari tiga bersaudara. Penulis menyelesaikan pendidikan dasar di SD Negeri 2 Kadipaten kemudian melanjutkan ke jenjang pendidikan menengah di SMP Negeri 1 Ponorogo dan dilanjutkan kembali ke jenjang pendidikan atas di SMA Negeri 2 Ponorogo. Pada jenjang perguruan tinggi penulis memulai pendidikan pada jenjang pendidikan Diploma di D3 Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya pada tahun 2013. Kemudian penulis menyelesaikan masa pendidikan diploma pada tahun 2016. Penulis kemudian melanjutkan kembali jenjang pendidikan sarjana di S1 Teknik Elektro ITS pada tahun 2016. Semasa kuliah pada jenjang diploma maupun sarjana penulis pernah melaksanakan kerja praktek di PT PLN dan juga PT LEN Industri. Penulis juga pernah menjadi asisten laboratorium elektronika dasar pada jenjang diploma, selain itu penulis juga aktif dalam unit kegiatan mahasiswa IBC (ITS Badminton Club). Beberapa penghargaan dibidang ekstrakurikuler berhasil didapatkan ditingkat Institusi.

Email : wahyuprayogo29@gmail.com
HP : 085745559605

-----Halaman ini sengaja dikosongkan-----