CHAPTER 4
Analysis and Design

4.1 Analysis

4.1.1 Arduino

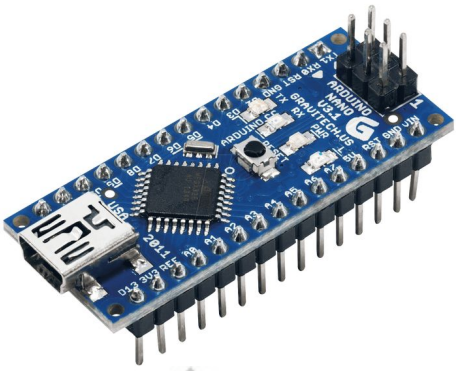
Arduino is an open source single-board microcontroller brought down from the wiring stage, intended for usability hardware in different fields. Equipment in arduino has a processor Atmel AVR and utilizes its own product and language.

4.1.2 Hardware

Equipment in Arduino has a few kinds, which enjoy benefits and drawbacks of each board. Arduino type utilization changed in accordance with the necessities, this will influence the sort of processor utilized. The more mind boggling the plan and program are made, it should likewise coordinate with the sort of regulator utilized. What distinguishes arduino with one another is the expansion of capacities in each board and the kind of microcontroller utilized. In this last undertaking, the kind of Arduino utilized is Arduino Nano.

4.1.3 Arduino Nano

Arduino is a stage for actual processing that is open source. Arduino isn't only an improvement apparatus, yet is a mix of equipment, programming language and incorporated Progressed Advancement Climate (IDE) An IDE is a piece of programming that job is to compose programs, assemble it into parallel code and transfer it to the memory of the microcontroller.



Picture 4.1.3 Arduino Nano

4.1.4 Arduino Nano Pin Configuration

Arduino Nano pin arrangement. Arduino Nano has 30 Pins. Here's the Arduino Nano pin setup.

1. The VCC is a pin that fills in as the force supply input pin advances.

2. GND is a ground pin for computerized power supply

3. AREF is the voltage reference for simple information sources. Utilized with the analogReference() work.

4. RESET is a LOW line that is utilized to reset (restart) the microcontroller. Generally utilized for included reset button safeguard impeding board Arduino principle.

5. Sequential RX (0) is the pin for getting sequential TTL information.

6. Sequential TX (1) is the pin for sending sequential information TT.

7. Outer Intrude on (Outside Interfere) is a pin that can be designed to trigger a hinder at a low worth, increments or diminishes, or changes in value.

8. The 8 Digit PWM yield is a pin that works for information analogWrite().

9. SPI is a pin that fills in as a correspondence support.

10. Driven is a pin that fills in as a pin that is set to esteem HIGH, then, at that point the LED will illuminate, when the pin is set to LOW then, at that point Opened. LEDs Available underlying on the Arduino Nano board.

11. Simple data sources (A0-A7) are pins that capacity as pins that can be measured/set from Ground up to 5 Volts, also permits to change the most elevated reach point or least they utilize the analogReference() work.

Picture 4.1.4 Arduino Nano Pin Configuration

Table 4.1.4 Arduino Nano Pin Configuration

Arduino Nano Pin Number	Arduino Pin Name
1	Digital Pin 0 (TX)
2	Digital Pin 0 (RX)
3 & 28	Reset
4 & 29	GND
5	Digital Pin 2
6	Digital Pin 3 (PWM)
7	Digital Pin 4
8	Digital Pin 5 (PWM)
9	Digital Pin 6 (PWM)
10	Digital Pin 7
11	Digital Pin 8
12	Digital Pin 9 (PWM)
13	Digital Pin 10 (PWM-SS)
14	Digital Pin 11 (PWM-MOSI)
15	Digital Pin 12 (MISO)
16	Digital Pin 13 (SCK)
18	AREF
19	Analog Input 0
20	Analog Input 1
21	Analog Input 2
22	Analog Input 3
23	Analog Input 4
24	Analog Input 5
25	Analog Input 6
26	Analog Input 7
27	VCC
30	Vin

4.1.5 Arduino Nano Specifications

The following are the specifications of the Arduino Nano:

1. Microcontroller chip using ATmega328p or Atmega168.

2. The operating voltage is 5 volts.

3. Input voltage (recommended) 7volt – 12 volt

4. There are 14 digital I/O pins and 6 of them as output PWM.

5. 8 Pin Input Analog.

6. 40 Ma Arus DC per pin I/O

7. Flash Memory 16 KB (Atmega168) or 32KB (Atmega328) 2KB used by the Bootloader.

8. 1 KbyteSRAM (Atmega168) or 2 Kbyte 32KB (Atmega328)

9. 512 Byte EEPROM (Atmega168) atau 1 Kbyte (Atmega328).

10. 16MHz Clock Speed.

11. Dimensity 1.85cm x 4.3cm.

4.1.6 Arduino Resources

The Arduino Nano can be controlled by means of a Smaller than usual B USB association, or through outer force supply with unregulated voltage between 6-20 Volts associated by means of pin 30 or pin VIN, or through the force supply outside with a directed voltage of 5 volts by means of pin 27 or pin 5V. The force source will be naturally chosen from the overvoltage source high. The FTDI FT232RL chip on the Arduino Nano will be dynamic when gets power through USB, when the Arduino Nano is fueled from an external perspective (Non-USB) then, at that point the FTDI Chip isn't dynamic and the 3.3V pin isn't accessible (doesn't yield voltage), while the TX and RX LEDs are additionally accessible glimmers when advanced pins 0 and 1 are in the Elevated place.

4.1.7 Arduino Nano Memory

Arduino nano utilizes the Atmega 168 microcontroller outfitted with a glimmer memory of 16 kbyte and can be utilized to store the fundamental program code. This blaze memory has been utilized 2 kbyte for the bootloader program while the Atmega328 is furnished with 32 kbyte streak memory and diminished by 2 kbyte for bootloader.

As well as being outfitted with streak memory, a microcontroller The ATmega168 and ATmega328 are additionally furnished with SRAM and EEPROM. SRAM and EEPROM can be utilized to store information while the primary program is running. SRAM size for ATmega168 is 1 kb and for ATmega328 is 2 kb while the EEPROM for ATmega168 is 512 b and for ATmega328 is 1 kb.

4.1.8 ATMega328 Microcontroller

Microcontroller is a chip that has capacities as an electronic circuit regulator and can by and large store programs in it. The microcontroller for the most part comprises a computer processor (Focal Preparing Unit), memory, certain I/O and supporting units like Simple-to-Advanced Converter (ADC) which is as of now coordinated in it.

Picture 4.1.8 ATMega328 Microcontroller Physical Form

ATMega328 microcontroller is a microcontroller yield from atmel which has a RISC (Diminished Guidance Set PC) engineering which each information execution measure is quicker than the CISC design (Finished Guidance Set PC).

This microcontroller has a few highlights, including :

1. 130 sorts of directions, practically completely executed in one clock cycle.

2. 32 x 8-cycle broadly useful registers.

3. Rates up to 16 MIPS with a clock of 16 Mhz.

4. 32 KB Streak memory and on arduino has a bootloader that utilizes 2 KB of flash memory as bootloader.

5. Have EEPROM (Electrically Erasable Programmable Read OnlyMemory) of 1KB as a semi-information stockpiling region lasting in light of the fact that EEPROM can in any case store information despite the fact that the force supply is killed.

6. Has SRAM (Static Irregular Access Memory) of 2KB.

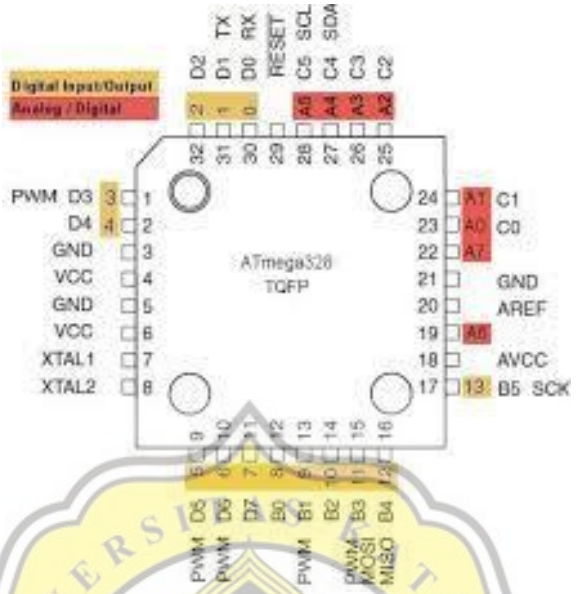
7. It has 14 computerized I/O pins, 6 of which are PWM (Heartbeat Width Modulation) yield.

8. Expert/Slave SPI Sequential interface.

The ATmega 328 microcontroller adopts Harvard architecture, which separates program code memory and data memory to maximize work and parallelism. Instructions in program memory are executed in one path. When an instruction is executed, the next instruction has been fetched from program memory.

This concept allows instructions to be executed in every clock cycle. Common 32 x 8 bit registers are used to support operations on the ALU (arithmetic logic unit), which can be executed in one cycle. Six of these general-purpose registers can be used as three 16-bit pointer registers in indirect addressing mode to retrieve data in the data memory space.

These three 16-bit pointer registers are called the X register (a combination of R26 and R27), the Y register (a combination of R28 and R29) and the Z register (a combination of R30 and R31). Most AVR commands are in 16-bit format. Each program memory address consists of a 16-bit or 32-bit instruction.



Picture 4.1.8 ATMega 328 Pin Configuration

Table 4.1.8 Port B Special Functions

Port	Function
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2) PCINT7 (Pin Change Interrupt 7)
PB6	XTAL1 (Chip Clock Oscillator pin 1) TOSC1 (Timer Oscillator pin 1) PCINT6 (Pin Change Interrupt 6)
PB5	SCK (SPI Bus Master Clock Input) PCINT5 (Pin Change Interrupt 5)
PB4	MISO (SPI Bus Master Input) PCINT4 (Pin Change Interrupt 4)

PB3	MOSI (SPI Bus Master Output) OC2A (Timer/Counter2 Output) PCINT3 (Pin Change Interrupt 3)
PB2	SS (SPI Bus Master Slave Select) OC1B (Timer/Counter1 Output) PCINT2 (Pin Change Interrupt 2)
PB1	OC1A (Timer/Counter1 Output Compare A) PCINT1 (Pin Change Interrupt 1)
PB0	ICP1 (Timer/Counter1 Input Capture Input) CLKO (Divided Sistem Clock Output) PCINT0 (Pin Change Interrupt 0)

Table 4.1.8 Port C Special Functions

Port	Function
PC 6	RESET(Reset Pin) PCINT14 (Pin Change Interrupt 14)
PC 5	ADC5(ADC Input Channel 5) SCL(2-Wire Serial Bus Clock line) PCINT13 (Pin Change Interrupt 13)
PC 4	ADC4(ADC Input Channel 4) SDA(2-Wire Serial Bus Clock line) PCINT12 (Pin Change Interrupt 12)
PC 3	ADC3(ADC Input Channel 3) PCINT11 (Pin Change Interrupt 11)

PC 2	ADC2(ADC Input Channel 3) PCINT10 (Pin Change Interrupt 10)
PC 1	ADC1(ADC Input Channel 1) PCINT9 (Pin Change Interrupt 9)
PC 0	ADC0(ADC Input Channel 0) PCINT8(Pin Change Interrupt 8)

Table 4.1.8 Port D Special Functions

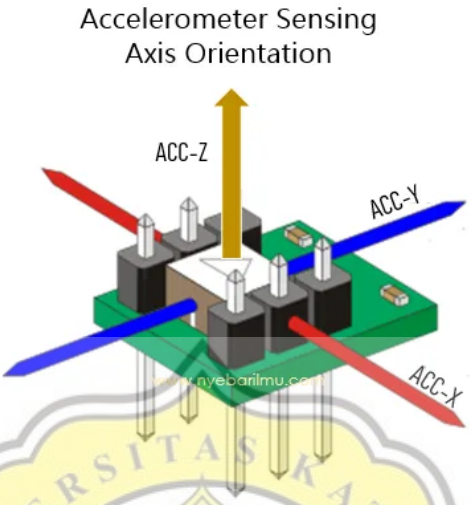
Port	Function
PD7	AIN1(Analog Comparator Negative Input) PCINT23 (Pin Change Interrupt 23)
PD6	AIN0(Analog Comparator Positive Input) OC0A (Timer0 Output compare Match B) PRINT22(Pin Change Interrupt 22)
PD5	T1(Timer 1 External Counter Input) OC0B (Timer0 Output compare Match B) PCINT21 (Pin Change Interrupt 21)
PD4	XCK (USART External Clock Input/Output) T0(Timer 0 External Counter Input) PCINT20 (Pin Change Interrupt 20)

PD3	INT1 (External Interrupt 1 Input) OC2B (Timer2 Output compare Match B) PCINT19 (Pin Change Interrupt 19)
PD2	INT0 (External Interrupt 0 Input) PCINT18 (Pin Change Interrupt 18)
PD1	TXD (USART Output Pin) PCINT17(Pin Change Interrupt 17)
PD0	RXD (USART Input Pin) PCINT16 (Pin Change Interrupt 16)

4.1.9 Arduino MPU6050

The MPU6050 module is a sensor that functions as an accelerometer as well as a gyroscope which is packaged into 1 module that is compatible with Arduino.

4.2.0 Accelerometer

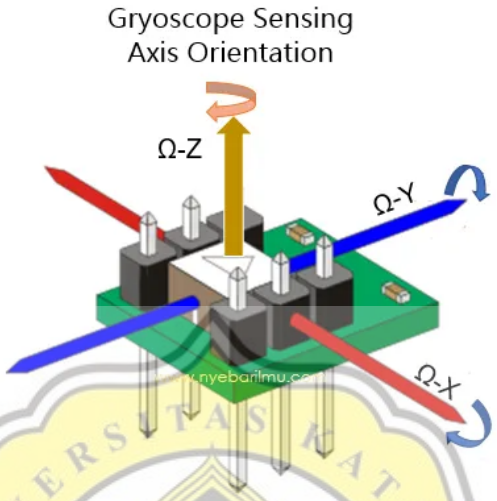


Accelerometer is a sensor used to quantify the speed of at least one item. The accelerometer can quantify both dynamic and static speed increase. Dynamic estimation is an estimation of the speed increase of a moving article, while static estimation is an estimation of the world's gravity. For instance, estimating vibrations that happen in vehicles, structures, and machines. Moreover, it can likewise be utilized to gauge vibrations, motor vibrations, dynamic distances and paces that happen inside the earth, with or without the impact of earth's gravity.

For this issue the accelerometer is all the more regularly used to gauge the point of tendency. The functioning rule of the accelerometer is essentially the guideline of utilizing speed increase (speed increase).

An illustration of an accelerometer in regular day to day existence is the jitter control work on a mobile phone.

4.2.1 Gyroscope



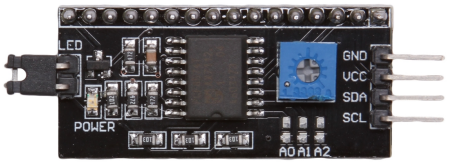
Gyroscope is a sensor that serves to determine the orientation of motion by resting on a wheel that rotates rapidly on an axis based on angular momentum. Gyroscope is a sensor device used to track the rotation or rotation of a device based on motion. In other words, a gyroscope is also referred to as a device that is used to maintain the orientation of an angle to remain stable. In activities such as tilting or rotating the phone, the gyroscope cannot work alone but with the help of the accelerometer. The most common example of using a gyroscope on a smartphone is the Google Sky Map application. You can also find it while enjoying 3D-based virtual reality content on a smartphone. In addition to mobile phones, gyroscopes are also used in other sophisticated devices such as drones and robots.

The gyroscope has an output in the form of angular velocity from the direction of 3 axes, namely, the x axis which will later become the phi angle (right and left) from the y axis which will later become the theta angle (top and bottom), and the z axis will later become the psi angle (front and back).

In essence, the gyroscope will detect the gravitational movement of the user who is doing the rotation of the head or while walking. Generally, the gyroscope will look for the orientation of the motion that has a focus on the disk rotating rapidly about the axis. This is done by a tool contained in the gyroscope called a gyro sensor.

4.2.2 I2C Module

Inter Integrated Circuit or often called I2C is a communication standard bidirectional serial using two specially designed channels to transmit or receive data. The I2C system consists of SCL (Serial Clock) and channels SDA (Serial Data) which carries data information between I2C and the controller. Devices connected to the I2C Bus system can operate as Master and Slave. The master is the starting device to transfer data on the I2C Bus by establishing a Start signal, ending the transfer data by generating a Stop signal, and generating a clock signal. Slave is the master-addressed device.



Pictures 4.2.2 Front View of I2c Module

4.2.3 Main features of I2C

The essential features of the I2C transport are according to the accompanying:

1. It simply incorporates two connections, to be explicit the consecutive data line (hereinafter called SDA) and successive clockline (hereinafter suggested as SCL).

2. Each IC related with I2C has another area novel programming accessible with expert/slave clear show, prepared to oblige various managers.

3. I2C is a successive vehicle with data bearing of 8 pieces (bytes), 2-way correspondence, with data move rates up to 100 Kbit/s in standard mode and 3.4 Mbit/s in mode fast.

4. The amount of ICs that can be related with the I2C transport is simply confined by the stack capacitance on the vehicle i.e most prominent 400pF.

4.2.4 Advantage I2C

The benefits of utilizing I2C include:

1. Limit the way of the connection between ICs.

2. Save the space of the PCB utilized.

3. Make a framework that is intended to be programming focused (effortlessly extended and upgraded).Creating the planned framework becomes standard, so it very well may be associated with the framework others additionally utilize the I2C transport.

4.2.5 Buzzer

Signal is an electronic part that capacities to change over electrical vibrations into sound vibrations. Fundamentally the functioning guideline of the ringer practically equivalent to an amplifier, so the bell additionally comprises of a loop that mounted on the stomach and afterward the curl is invigorated so that turns into an electromagnet, the loop will be drawn in or out, contingent upon the bearing of the current and the extremity of the magnet, on the grounds that the loop is mounted on stomach then every development of the loop will move the stomach to and fro to cause the air to vibrate which will create sound. The signal is normally utilized as a pointer that the cycle has been finished or a mistake happened in a gadget (alert).

Pictures 4.2.5 Buzzer

4.2.6 Jumper Cable

4.2.6.1 Definition Jumper Cable

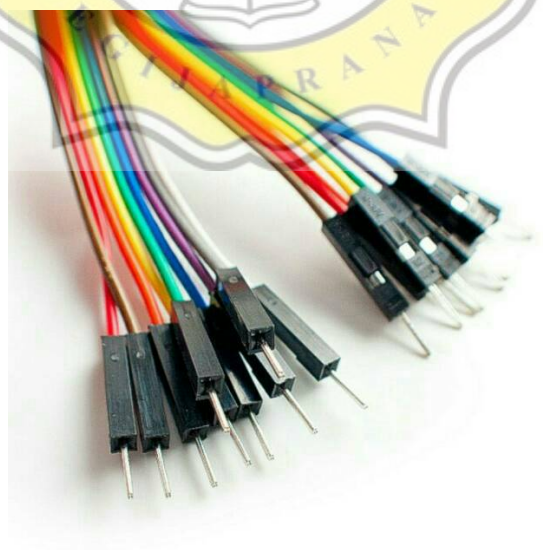
Jumper link is a term for a little measurement link which in the realm of gadgets is utilized to interface at least two focuses and can likewise associate 2 segments of hardware.

4.2.6.2 Jumper Cable Type

There are several types of jumper cables which are distinguished by cable connectors, namely:

Male - male

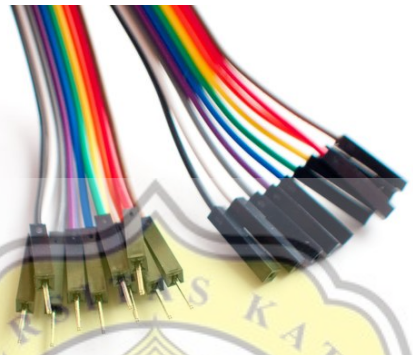
This type of jumper cable is used for male to connect male on both ends of the cable.



Pictures 4.2.6.2 Male to male jumper cable

Male – female

This type of jumper cable is used for male to connect females with one end of the cable connected to the male and one the other end with a female connection.



Pictures 4.2.6.2 Male to Female jumper cable

Female - female

This type of jumper cable is used for females to connect females on both ends of the cable.

Pictures 4.2.6.2 Female to Female jumper cable

4.2.7 Switch

Switch is an electronic gadget that is utilized to disengage the force matrix or to interface with it. So switch on Fundamentally it's anything but an interfacing gadget or an electric electrical switch. Notwithstanding network solid flows, little switches are additionally utilized for segment apparatuses and feeble current hardware.

Pictures 4.2.7 Switch

In straightforward terms, a switch comprises two metal edges appended to a circuit, and can be associated or isolated by the association state (on) or separated (off) in the circuit. General association contact material chose to be consumption safe. Assuming the metal utilized is made of normal oxide material, the switch will regularly not work. To lessen the impact For this erosion, at any rate the contact metal should be plated with hostile to consumption metal and

against rust. Essentially fastens can be applied to mechanical sensors, since it very well may be utilized as an aide on the microcontroller for gadget settings in charge.

4.2.8 Battery 9V

A battery is a gadget that can change over put away synthetic energy into electrical energy that can be utilized by an electronic gadget. Practically all compact electronic gadgets like cell phones, PCs, spotlights, or controllers use batteries as their force source. With the battery, we don't have to associate an electric link to have the option to actuate our electronic gadgets so they can be handily conveyed all over the place. In our day by day life, we can discover two sorts of batteries, in particular batteries that must be utilized once (single use) and batteries that can be re-energized (battery-powered).

Pictures 4.2.8 Battery 9V

4.2.9 Arduino Programming

Arduino was made for amateurs despite the fact that it doesn't have a fundamental programming language at all since it utilizes C++ language that has been streamlined through libraries. Arduino utilizes Handling programming used to compose programs into Arduino.

Preparing itself is a mix of C++ and Java. This Arduino programming can be introduced on different working frameworks (operating system, for example, LINUX, Macintosh operating system, Windows. IDEArduino programming comprises of 3 (three) sections:

1. Program manager, for composing and altering programs in handling. Posting programs on Arduino are called draws.

2. Compiler, a module that capacities to change the preparing language (program code) into double code since parallel code is one the solitary programming language comprehended by the microcontroller.

3. Uploader, a module that capacities to enter twofold code into the microcontroller memory. Order structure on arduino in line enormous comprises 2 (two) sections, specifically void arrangement and void circle. Void arrangement contains orders to be executed just a single time since Arduino is turned on, while the void circle contains orders that will execute over and over as long as the Arduino is fueled on.

Pictures 4.2.9 Arduino Software Logo

In the Arduino programming language there are three fundamental parts, in particular the structure,variables and capacities:

1. Arduino Program Construction The fundamental design of the Arduino programming language is basic and simple. All together for the program to run appropriately, it is important at any rate two sections or capacities for example arrangement() which is called just one time, typically for program instruction (input settings or sequential settings, and others). Also, circle() is a spot to execute a program straightforwardly over and again, as a rule to understand info or trigger yield. Here is the type of composing:

```
Void setup()
{
//Statement;
}
Void loop()
{
//Statement;
}
```

Setup()

The arrangement() work is just called once during the program begin strolling. The arrangement() work is valuable for instating pin mode or starting sequential correspondence. This arrangement() should exist albeit no program will be executed. Here's the shape composing:

```
Void setup()
{
```

```
pinMode(led, OUTPUT); //set led sbg
output  }
Void loop()
{
//statement;
                                }
```

Loop()

Subsequent to setting up the instruction on arrangement(), the accompanying makes a circle() work. As the name suggests, this capacity will rehash the current program persistently, so the program will change and react to include. This circle() capacity will be a functioning control arduino board.

An example of using the loop() function is as follows:

```
Void setup()
{
pinMode(led, OUTPUT); //set led sbg output
}
Void loop()  {
digitalWrite(led, HIGH); //set led on  delay (500); /
tunda untuk ½ detik digitalWrite(led, LOW); //set led
off  delay (500); //tunda untuk ½ detik
}
```

2. Variable

This variable serves to oblige the worth of numbers and give the name as per the need to make the program. With utilizing a variable, then, at that point the current worth can be changed openly. A variable should be proclaimed first, and it very well may be utilized as a placeholder for the information peruser to be put away or given a value early.

3. Functions

The elements of Arduino programming comprise of:

Computerized I/O Capacities

There are three capacities for computerized I/O, in particular pinMode(pin, mode), digitalWrite(pin, esteem), and int digitalRead(pin).

Simple I/O Capacity

There are three capacities for simple I/O, in particular analogReference(type), int analogRead(pin), and analogWrite(pin,value)-PWM.

Time Capacity

The time work comprises an unsigned long mailing list (), delay(ms) dandelayMicrosonds(us).

Math Capacities

Numerical capacities consist of min(x,y), max(x,y), abs(x), sqrt(x) furthermore, pow(base, dramatic).

4.3 Algorithm

Utilization of separating techniques on information sensor yield is very significant, in light of the fact that there is clamor or obstruction that can lessen the exactness of the yield worth of sensors. Along these lines, utilizing the information sifting strategy metode required as an answer for the issue. There are a few sorts of channel strategies which can be applied to the sensor yield worth, for example, the Kalman Channel technique, Correlative Inner channels and channels of the sensor relying upon the sensor type. AlthoughThere are a few sorts of information channel strategies for various sensors, the objective remaining parts as before, is to diminish or limit commotion. Reciprocal channel is one of the channel strategies with unimplemented intricacy since it doesn't need a great deal of factors in execution, a couple of factors like alpha (channel coefficient), examining time,. Worth tendency point of the spinner and accelerometer.

Our first step is to collect data from the accelerometer and gyroscope sensors when the elderly perform their normal activities. Then don't forget to test the data when the elderly are lying on their stomach, tilted left and right side. After that, we create a scenario of falling several times, which will later take the upper threshold value (DHPF) data and also take the lower threshold value (DLPF) which can later be entered into the complementary filter formula which aims to obtain more accurate data by reducing noise. So that the data can be more accurate in deciding whether the elderly actually fell or not. So that the buzzer can provide valid information to the people around. Here is an explanation of this formula:

CF = (data 1 * A) + (data 2 * B)

(A + B)

Data 1 & Data 2 = Accelerometer

A = load (0.95)

B = load (0.05)

Note = A + B = 1

Table 4.1 Analysis Data 1

Position	X	Y	Z
Berbaring	16300	-540	522
Tengkurap	-15564	-332	506
Miring Kiri	2080	-16192	416
Miring Kanan	3352	15903	313

Table 4.2 Analysis Data 2

Position	X	Y	Z
Berbaring	16076	-660	-445
Tengkurap	-15834	-378	607
Miring Kiri	-6014	-15372	588
Miring Kanan	6736	14024	390

Table Analysis Data 3

Position	X	Y	Z
Berbaring	16920	-540	506
Tengkurap	-16256	963	445
Miring Kiri	6868	-15848	381
Miring Kanan	6020	15072	200

4.2 Design

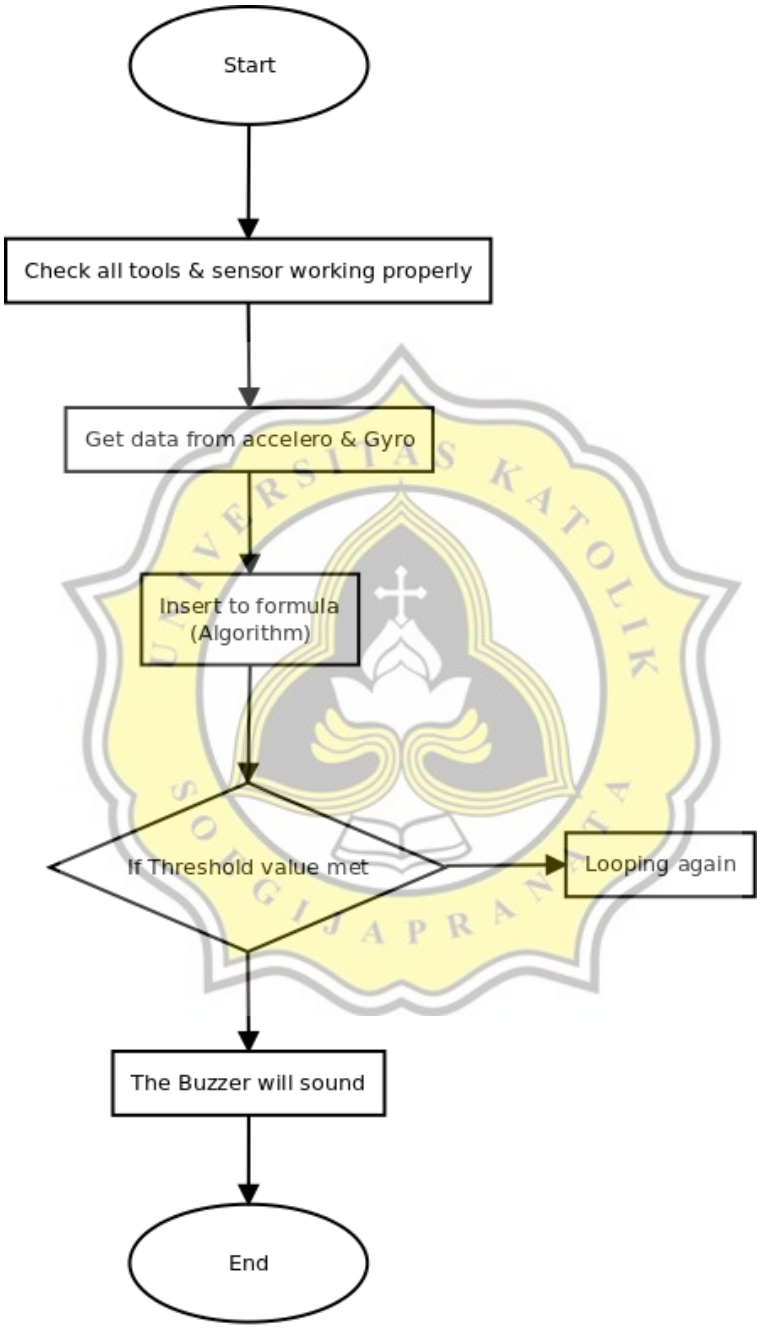


Illustration Flowchart 4

This flowchart more or less explains a lot at least about how this program works. First we start and upload our program that we will run. If when checking all the tools and sensors are running perfectly, the buzzer will give a sound signal for about 3 seconds. Then the tool can be used immediately. Initially we collect data from accelerometer and gyroscope sensors, after collecting data we create a reasonable fall scenario in the elderly, which can later be used as upper and lower threshold values. After that the program will enter into the algorithm formula that we have created. If one of the upper/lower threshold values is met, the buzzer will sound to alert people around that an elderly person has fallen. If the value is not met, the buzzer will not sound and the program will loop again to read the latest data received by the accelero sensor and gyroscope.

Tool Design