

Design of Measure Angle Tools as a Learning Tool for Learning Angle Measurement Precision Approach Path Indicator

Bayu Purbo Wartoyo
Aviation Polytechnic of Makassar
E-mail: byoe1234@gmail.com

Abstract: At present, the Makassar Aviation Polytechnic does not yet have a vertical angle measurement tool. This tool is needed to support the learning activities of cadets at the Makassar Aviation Polytechnic. This research is a form of engineering research making the measure angle tool as a learning aid. This research produces a vertical angle measuring instrument that can measure the angle of the precision approach path indicator (PAPI) in the aviation engineering department of the Makassar Aviation Polytechnic. The use of a tool that makes it easier for technicians is that it uses an LCD display that can directly display the PAPI tilt angle value. The components used in the manufacture of the Measure Angle tool are available in the market and are affordable so that if there is the damage it will be efficient during maintenance. This research produces a product in the form of microcontroller-based interactive learning media that can be used by cadets in the Makassar Aviation Polytechnic environment.

Keywords: Tool, Measure Angle, Aviation Polytechnic, PAPI

Date of Submission: 10-11-2020

Date of acceptance: 26-11-2020

I. Introduction

Aviation Polytechnic of Makassar is a Technical Implementation Unit (TIU) in the field of air transportation education and training within the Air Transportation Human Resources Development Agency 'BPSDM' which aims to create useful people for the world of aviation.

Makassar Aviation Polytechnic does not yet have a vertical angle measurement tool. This angle measurement tool is needed to support cadets' practicum learning activities. In addition to knowing the angle of the precision path indicator, this tool is also useful in training to simulate a precision approach path indicator calibration. Therefore, cadets, have difficulty knowing the angle value when simulating measurements and determining the angle of the precision path indicator.

Based on the above, the researcher tries to design a microcontroller-based tool to measure the vertical angle and the angle of similarity to the Precision Approach Path Indicator (PAPI).

II. The Methods

This research is a type of engineering research carried out from August 2020 to September 2020 at the Makassar Aviation Polytechnic Laboratory. The research stages are as follows:

1. The design stage of the vertical angle measurement system that will be used for measuring the angle of the precision approach path indicator
2. The assembly stage, assembling a vertical angle measurement system that will be used for measuring the precision approach path indicator angle
3. The trial stage, which is to test the system that has been made whether it is running properly

Model design: the tool made has a working principle of measuring a vertical or slope starting from the horizontal direction. The movement of the tool is set by moving clockwise with the positive sign (+ or down), and moving counterclockwise with the negative sign (- or up)

Create a wiring diagram for the Measure Angle tool

The wiring diagram is intended to make it easier to assemble the Measure Angle tool in which there is a path to connect one component to another so that it can work according to the instructions given.

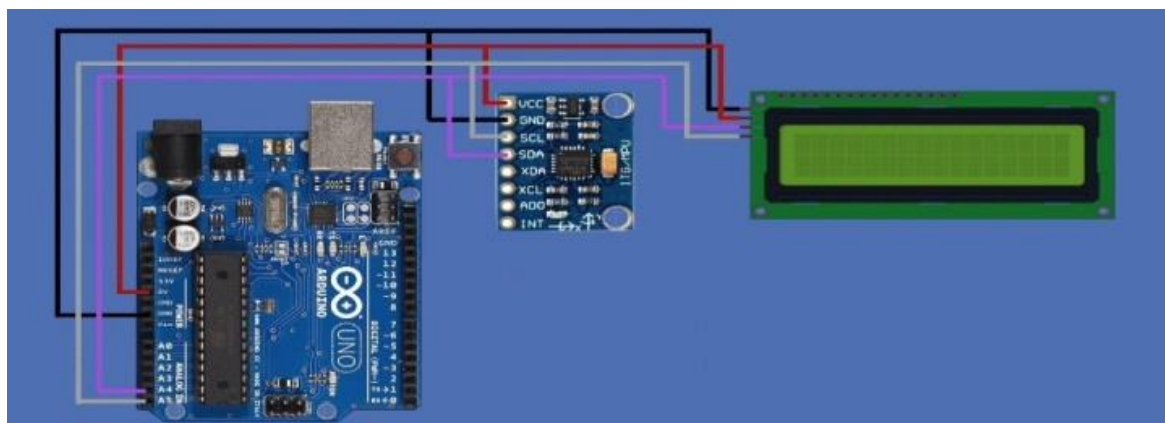


Figure 1. Design of the vertical angle measure

The components of the angle measure tool

The main components in making the Measure Angle tool, namely, Arduino, gyroscope sensors MPU 6050, type g-51 9250, I2C LCD, red laser and 20x4 LCD display, then as for complementary accessories, namely, a tripod for tool mounts when used to measure angles, frame, spicer bolts, single male header (fishbone), solder, tinol (tin solder), and male-male and male-female jumper cables.

How the System Works

To perform the calibration and angle measurement, the following steps are taken.

1. Place the Mockup Precision Approach Path Indicator on a flat spot.
2. Calibrate the mockup so that the value is 0° .
3. Then for the angle calibration exercise, it can be obtained by turning the existing PAPI mockup legs so that the desired angle results are obtained.

System Testing

System testing is done by comparing the system made with the original system in the field, is there a significant difference or can it approach the existing system in the field, so that the equipment made will be able to present the existing tools in the field later?

III. The Results

Hardware

Hardware that has been successfully made in this study is a microcontroller-based Measuring Angel tool. This hardware consists of Arduino Uno, MPU/I56 Gyroscope Sensor, LCD 16x2, I2C Sensor, Jumper Cables, 5 volt DC Supply. The explanation of the results of making hardware devices is as follows:

a. Arduino Uno

Arduino is an open-source microcontroller board based on the ATmega338P microcontroller and developed by Arduino CC. The board is equipped with a set of digital and analog input / output pins that can be connected to various expansion boards and other circuits. In this study, the microcontroller used is the Arduino Uno which functions to program the MPU/I56 Gyroscope sensor and display it on a 16x2 LCD.



Figure 2.Arduino Uno

Power supply

The power supply or commonly called the Power Supply is a device that supplies electricity to one or more electrical loads. The power supply is an important part in electronics which functions as a source of electricity, for example in batteries or batteries.



Figure 3.Power Supply 5VDC

In the power supply, the adapter is used to convert the AC220V mains voltage to DC voltage. In this study, the expected output voltage for the adapter is 5VDC.

MPU/156 sensor

The MPU/156 sensor functions as an angle tilt reader based on data from the accelerometer sensor and the gyroscope sensor. This sensor is also equipped with a temperature sensor that can be used to measure the ambient temperature. The data path used on this sensor is the I2C data line.

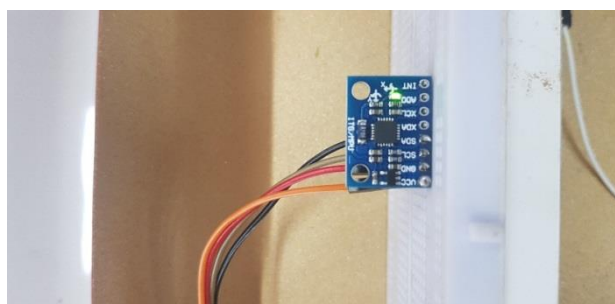


Figure 4. MPU/156 sensor

The MPU/156 sensor is connected with pins A4 and A5 so that Serial Clock (SCL) and Serial Data (SDA) data can send/receive information/data serially using the data line and also wait for the reading of commands from the master whether to accept/write data or send to the master. The VCC and GND pins function to connect a voltage of 5VDC and Ground.

LCD16x2

Liquid Crystal Display (LCD) is a type of display media that uses liquid crystals as the main display. LCD has been used in various fields such as electronic devices such as televisions, calculators, or computer screens.



Figure 5. Liquid Crystal Display (LCD)

This LCD functions to display the results of the MPU/156 sensor measurements and is set via Arduino so that it can display readings on the LCD 16x2. In addition, it is also for editing what you want to display on the LCD.

I2C (Inter Integrated Circuit)

Inter-Integrated Circuit or often referred to as I2C is a two-way serial communication standard using two channels specially designed to send or receive data. The I2C system consists of Serial Clock (SCL) and Serial Data (SDA) which carry data information between the I2C and the controller.



Figure 6. Inter Integrated Circuit

In this circuit, the pins on the 16x2 LCD are connected with I2C in order to facilitate the two-way serial communication. When the LCD and I2C have been connected, the pins on I2C themselves have 4 connectors. Among them are VCC as a voltage source which includes 5VDC and GND Pin as Grounding. PIN Serial Clock (SCL) and PIN Serial Data (SDA) are connected to PIN A4 and A5 on Arduino.

Mockup Precision Approach Path Indicator (PAPI)

Mockup made of the size is made as close as possible to the original equipment which uses material from an iron plate with a length of 60 cm, width 30 cm and height 25 cm, height 30 cm feet.



Figure 7. Mockup Precision Approach Path Indicator

Software

The software used to program the entire research is written in C using the Arduino software. The program made in this study is a setting on the MPU/156 sensor sensitivity and displays on the LCD.

a. Design Software on Arduino Uno

The Arduino Uno used in this study has an embedded boot loader

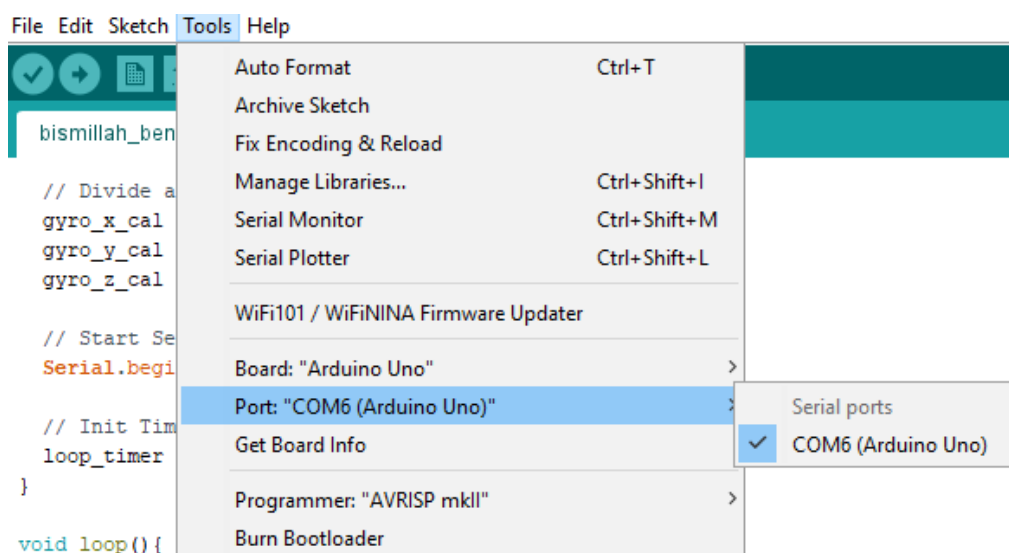
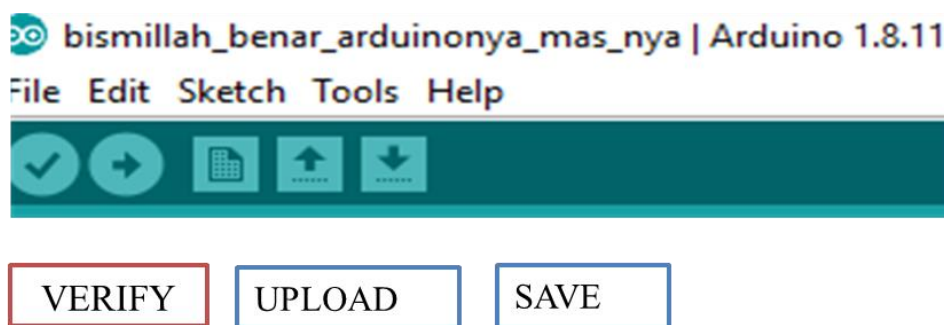


Figure 8. Choice of the serial port on Arduino software

To handle serial/USB communication with a computer, the serial port option used is the Arduino Uno board, which is on the Tools menu > Serial Port. The Serial Port on the Arduino Software occurs in the picture. The program that has been created can be verified by pressing the verify button, if there is no error in the program, then the program can be saved by pressing the save button and then uploading it is presented in the Arduino software as shown in Figure 4. The function of the Arduino software in making measurement tools Angel to program so you can see the result of the reading from the MPU/156 sensor.



The software that has been made in this study includes a program of duration of therapy and monitoring of pressure values on the Measuring Angel device.

b. MPU/156 Sensor Reading Program

The pressure value reading program in this study is needed because the output from the sensor must be set in order to get the required value. The pressure value readings are as follows:

This sketch requires another library, the New Liquid Crystal Library for I2C. This can be done by downloading the latest version of BitBucket. This is a ZIP file, so it is necessary to install to the Arduino IDE. Instead, the Arduino Wire library for I2C is used and the registers in the MPU/165 are directly manipulated.

In the initial sketch of Setup, the researcher calibrates the sensor by reading the acceleration data and the gyroscope 100 times. These are then averaged to arrive at a baseline. Due to this operation, the device will delay a few seconds before giving the first reading, the author is only interested in getting the Pitch and Roll value in degrees. Because the data was extracted too fast for the LCD screen to handle, the researcher used a counter and only took a reading for every 100 passes. After a few seconds of pause, the LCD screen will show you the angle, both pitch, and roll. The LEDs must be properly lit. Try moving the level around and pay attention to the LED reading and display.

A programming language is a command given to Arduino through a processing language as follows:

```
// Get data from MPU-6050
read_mpu_6050_data();

//Subtract the offset values from the raw gyro values
gyro_x -= gyro_x_cal; 3,00
gyro_y -= gyro_y_cal; 2,75
gyro_z -= gyro_z_cal; 3,00

//Gyro angle calculations. Note 0.0000611 = 1 / (250Hz x 65.5)

//Calculate the traveled pitch angle and add this to the angle pitch variable
angle_pitch += gyro_x * 0.0000611;
//Calculate the traveled roll angle and add this to the angle roll variable
//0.000001066 = 0.0000611 * (3.142(PI) / 180 deg) The Arduino sin function is in radians
angle_roll += gyro_y * 0.0000611;

//If the IMU has yawed transfer the roll angle to the pitch angle
angle_pitch += angle_roll * sin(gyro_z * 0.000001066);
//If the IMU has yawed transfer the pitch angle to the roll angle
angle_roll -= angle_pitch * sin(gyro_z * 0.000001066);

//Accelerometer angle calculations

//Calculate the total accelerometer vector
```

```

acc_total_vector=sqrt((acc_x*acc_x)+(acc_y*acc_y)+(acc_z*acc_z));

//57.296 = 1 / (3.142 / 180) The Arduino asin function is in radians
//Calculate the pitch angle
angle_pitch_acc = asin((float)acc_y/acc_total_vector)* 57.296;
//Calculate the roll angle
angle_roll_acc = asin((float)acc_x/acc_total_vector)* -57.296;

//Accelerometer calibration value for pitch
angle_pitch_acc -= 3.0;
//Accelerometer calibration value for roll
angle_roll_acc -= 3.0;

if(set_gyro_angles){

//If the IMU has been running
//Correct the drift of the gyro pitch angle with the accelerometer pitch angle
angle_pitch = angle_pitch * 0.9996 + angle_pitch_acc * 0.0004;
//Correct the drift of the gyro roll angle with the accelerometer roll angle
angle_roll = angle_roll * 0.9996 + angle_roll_acc * 0.0004;
}
else{
//IMU has just started
//Set the gyro pitch angle equal to the accelerometer pitch angle
angle_pitch = angle_pitch_acc;
//Set the gyro roll angle equal to the accelerometer roll angle
angle_roll = angle_roll_acc;
//Set the IMU started flag
set_gyro_angles = true;
}

//To dampen the pitch and roll angles a complementary filter is used
//Take 90% of the output pitch value and add 10% of the raw pitch value
angle_pitch_output = angle_pitch_output * 0.9 + angle_pitch * 0.1;
//Take 90% of the output roll value and add 10% of the raw roll value
angle_roll_output = angle_roll_output * 0.9 + angle_roll * 0.1;
//Wait until the loop_timer reaches 4000us (250Hz) before starting the next loop

//Print to Serial Monitor
//Serial.print(" | Angle = "); Serial.println(angle_pitch_output);

//Increment the display counter
displaycount = displaycount + 1;
}

File Edit Sketch Tools Help
Measure_Angle $
*/

// Include Wire Library for I2C
#include <Wire.h>

// Include NewLiquidCrystal Library for I2C
#include <LiquidCrystal_I2C.h>

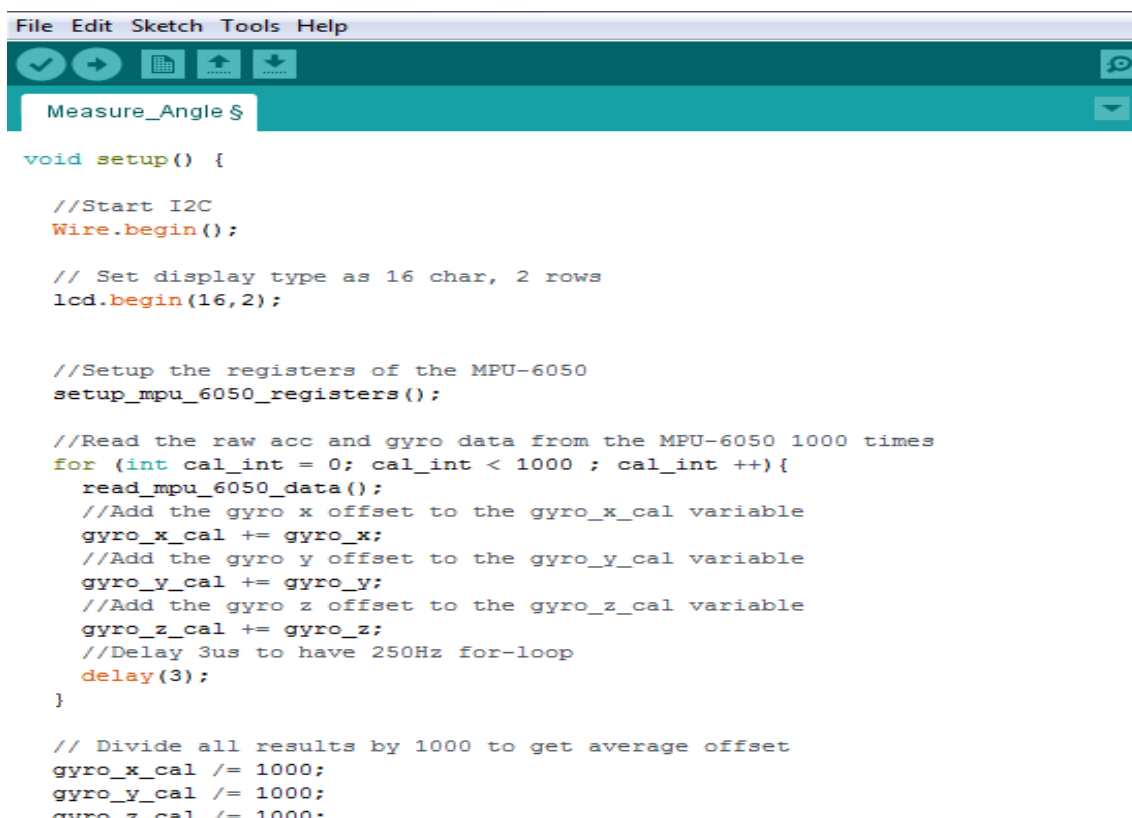
// Define LCD pinout
const int en = 2, rw = 1, rs = 0, d4 = 4, d5 = 5, d6 = 6, d7 = 7, bl = 3;

// Define I2C Address - change if required
const int i2c_addr = 0x3F;

LiquidCrystal_I2C lcd(0x27, 20, 4);
// Level LEDs

```

Figure 8. Activation of the LCD display



```

File Edit Sketch Tools Help
Measure_Angle $
void setup() {
  //Start I2C
  Wire.begin();

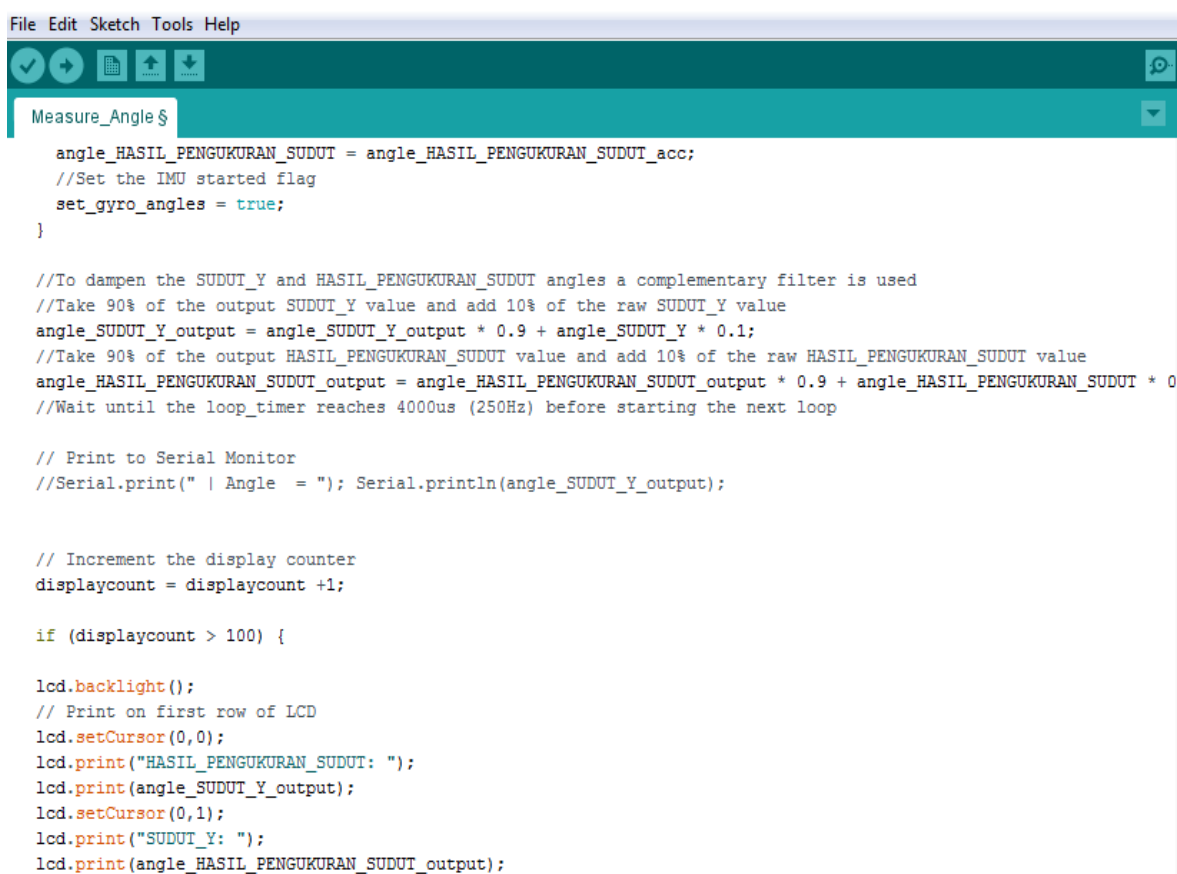
  // Set display type as 16 char, 2 rows
  lcd.begin(16,2);

  //Setup the registers of the MPU-6050
  setup_mpu_6050_registers();

  //Read the raw acc and gyro data from the MPU-6050 1000 times
  for (int cal_int = 0; cal_int < 1000 ; cal_int++){
    read_mpu_6050_data();
    //Add the gyro x offset to the gyro_x_cal variable
    gyro_x_cal += gyro_x;
    //Add the gyro y offset to the gyro_y_cal variable
    gyro_y_cal += gyro_y;
    //Add the gyro z offset to the gyro_z_cal variable
    gyro_z_cal += gyro_z;
    //Delay 3us to have 250Hz for-loop
    delay(3);
  }

  // Divide all results by 1000 to get average offset
  gyro_x_cal /= 1000;
  gyro_y_cal /= 1000;
  gyro_z_cal /= 1000;
  
```

Figure 9. MPU-9250 registration



```

File Edit Sketch Tools Help
Measure_Angle $
  angle_HASIL_PENGUKURAN_SUDUT = angle_HASIL_PENGUKURAN_SUDUT_acc;
  //Set the IMU started flag
  set_gyro_angles = true;
}

//To dampen the SUDUT_Y and HASIL_PENGUKURAN_SUDUT angles a complementary filter is used
//Take 90% of the output SUDUT_Y value and add 10% of the raw SUDUT_Y value
angle_SUDUT_Y_output = angle_SUDUT_Y_output * 0.9 + angle_SUDUT_Y * 0.1;
//Take 90% of the output HASIL_PENGUKURAN_SUDUT value and add 10% of the raw HASIL_PENGUKURAN_SUDUT value
angle_HASIL_PENGUKURAN_SUDUT_output = angle_HASIL_PENGUKURAN_SUDUT_output * 0.9 + angle_HASIL_PENGUKURAN_SUDUT * 0.1;
//Wait until the loop_timer reaches 4000us (250Hz) before starting the next loop

// Print to Serial Monitor
//Serial.print(" | Angle = "); Serial.println(angle_SUDUT_Y_output);

// Increment the display counter
displaycount = displaycount +1;

if (displaycount > 100) {

  lcd.backlight();
  // Print on first row of LCD
  lcd.setCursor(0,0);
  lcd.print("HASIL_PENGUKURAN_SUDUT: ");
  lcd.print(angle_SUDUT_Y_output);
  lcd.setCursor(0,1);
  lcd.print("SUDUT_Y: ");
  lcd.print(angle_HASIL_PENGUKURAN_SUDUT_output);
  
```

Figure 10. Display angle reading caption


```
File Edit Sketch Tools Help
Measure_Angle $
void read_mpu_6050_data() {
  //Start communicating with the MPU-6050
  Wire.beginTransmission(0x68);
  //Send the requested starting register
  Wire.write(0x3B);
  //End the transmission
  Wire.endTransmission();
  //Request 14 bytes from the MPU-6050
  Wire.requestFrom(0x68, 14);
  //Wait until all the bytes are received
  while(Wire.available() < 14);

  //Following statements left shift 8 bits, then bitwise OR.
  //Turns two 8-bit values into one 16-bit value
  acc_x = Wire.read() << 8 | Wire.read();
  acc_y = Wire.read() << 8 | Wire.read();
  acc_z = Wire.read() << 8 | Wire.read();
  temp = Wire.read() << 8 | Wire.read();
  gyro_x = Wire.read() << 8 | Wire.read();
  gyro_y = Wire.read() << 8 | Wire.read();
  gyro_z = Wire.read() << 8 | Wire.read();
}
```

Figure 11. Final activation of the MPU-6050 gyroscope

IV. Discussion

Calibration and measurement of angles using the Precision Approach Path Indicator mockup.

To perform calibration and angle measurement, the following steps are carried out.

1. Place the Mockup Precision Approach Path Indicator on a flat place.

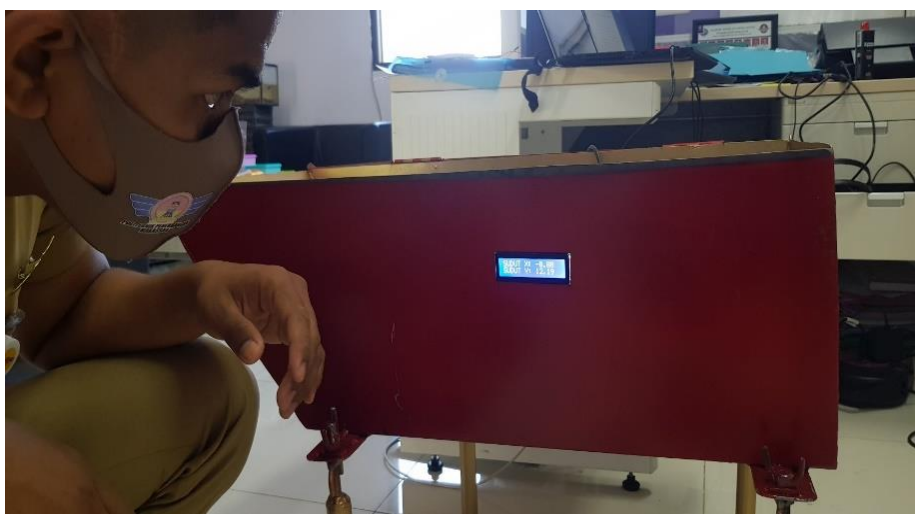


Figure 12. Laying the mockup on a flat place

1. Perform a mockup calibration so that the value is 0° .



Figure 13. Mockup Calibration to get a value of 0°

2. Then for the angle calibration exercise it can be obtained by rotating the existing PAPI mockup legs so that the desired angle results are obtained.



Figure 14. The precision calibration exercise of approach path indicator (PAPI)

3. Recording the results of the tilt angle listed on the LCD display of the Measure Angle tool and calculating the height of the PAPI light emitted on the object, then entered into the table for comparisons to determine the accuracy of the tool.



Figure 15. The measurement results are shown on the LCD of the Measure Angle tool

b. Comparison of Measurement Results.

To find out the accuracy of the Measure Angle tool, it is necessary to make a comparison with the test results using a clinometer and a PAPI mockup that is equipped with a measure angle tool, the data obtained are as follows:

Table 1. Comparison of Measurement Results

No Box PAPI 09	Measure Angle	ground inspection
Position 1	2.42 ⁰	2.45 ⁰
Position 2	2.78 ⁰	2.79 ⁰
Position 3	3.24 ⁰	3.26 ⁰
Position 4	2.53 ⁰	3.55 ⁰

V. Conclusion

This research produces a product in the form of microcontroller-based interactive learning media that can be used by cadets (i) in the Makassar Aviation Polytechnic environment. Based on the results of calculations with the results of flight calibration (flight inspection) as a benchmark or basis for the slope of the PAPI box, the results of the slope deviation measured by the clinometer with the Measure Angle are not much different, this shows that the Measure Angle tool is accurate enough to be used in carrying out measurements on land (ground inspection). For the Angle Measure tool, it is necessary to attach a water pass to find the angle 0⁰.

Acknowledgment

The author would like to thank Prof.Dr.-Ing.M. Yamin Jinca, a lecturer in the field of Transportation at Hasanuddin University as an examiner in this research seminar, as well as to the Director of the Makassar Aviation Polytechnic who has funded this research.

References and further reading

- [1]. Sugiyono, 2016. *Qualitative and Quantitative Research Methods R & D*. Bandung Alfabeta
- [2]. Regulation of the Minister of Transportation of the Republic of Indonesia, Number: KP 39, 2015, *Technical Standards and Operations of Civil Aviation Safety Regulations - Part 139 (Manual of Standard Casr - Part 139) volume 1, airports (aerodromes)*
- [3]. Latief, M. A. 2014. *Research Methods om Language Learning: An Introduction*. Malang: UM Press
- [4]. Regulation of the Minister of Transportation of the Republic of Indonesia, Number: PM 85 of 2014, *Guidelines and Procedures for the Implementation of Electronic Calibration and Flight Failure*
- [5]. Directorate General of Civil Aviation, Department of Transportation of the Republic of Indonesia, Number: SKEP 157 IX, 2003, *Guidelines for the Maintenance and Reporting of Aviation Electronics and Electrical Facilities Equipment*

Bayu Purbo Wartoyo. "Design of Measure Angle Tools as a Learning Tool for Learning Angle Measurement Precision Approach Path Indicator." *IOSR Journal of Research & Method in Education (IOSR-JRME)* , vol. 10, no. 6, 2020, pp. 23-33.