

PELATIHAN GALILEO

Getting Started With INTEL Galileo



DAFTAR ISI

| | |
|--|-----------|
| Daftar Isi..... | 2 |
| MODUL I. <i>Pengenalan Intel Galileo</i> | 3 |
| MODUL II. <i>Upload Program ke Intel Galileo</i>..... | 6 |
| MODUL III. <i>Digital Input Output</i>..... | 10 |
| MODUL IV. <i>Analog Input Output PWM</i>..... | 15 |
| MODUL V. <i>GPIO Over Linux</i> | 19 |

MODUL I

PENGENALAN INTEL GALILEO

Tujuan

- Mengetahui Intel Galileo
- Menyambungkan dan menyiapkan perangkat Intel Galileo pada Windows

Dasar Teori

Intel Galileo adalah board mikrokontroler yang berbasis prosesor Intel Quark SoC X1000. Prosesor ini merupakan prosesor 32-bit yang memang dirancang khusus untuk peralatan yang memerlukan daya rendah tetapi tetap memiliki kinerja yang tinggi.

Board Galileo merupakan modul mikrokontroler pertama kali di dunia yang menggunakan prosesor Intel dengan desain hardware yang kompatibel penuh dengan Arduino. Jika melihat dari desain boardnya, modul Galileo memiliki susunan pin yang sesuai dengan Arduino UNO. Oleh karena itu, modul Galileo juga bisa terhubung ke berbagai macam shield dari Arduino.

Sama seperti modul Arduino yang lain, modul Galileo juga memiliki pin digital (0-13) dan pin Analog (0-5), ICSP Header dan UART TTL pada pin 0 dan pin 1.

Perihal tegangan kerja, board Galileo dapat bekerja pada tegangan 3,3VDC dan 5VDC (ada konfigurasi jumper)

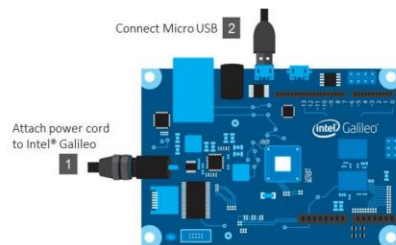
Fitur :

- Prosesor Quark SoC X1000 (400MHz, 512 SRAM)
- 8MByte untuk Firmware / bootloader
- 11KByte EEPROM
- microSD Card (up to 32GB)
- 10/100 Ethernet (RJ45)
- USB 2.0 Host and Device
- RS-232 Port
- mini PCI Express (mPCIe) socket
- SPI, I2C, UART, PWM

Board Galileo dapat program dengan menggunakan Arduino IDE. Selain itu, kita bisa memanggil program eksternal dengan menggunakan perintah "System()" di Arduino IDE, seperti program python, OpenCV ataupun program lain yang bisa jalan di Linux.

Langkah Praktikum

- Download IDE dan firmware files terbaru untuk intel galileo pada link berikut :
<https://communities.intel.com/community/makers/software/drivers>
- Ekstrak semua file yang sudah di download pada drive C:\
- Ekstrak file IDE “Intel Galileo Arduino SW 1.5.3 on Windows” ke Desktop.
- Nyalakan Intel galileo dengan menyambungkan kabel power.
- Sambungkan Intel Galileo dengan computer menggunakan kabel USB



- Buka device manager, maka akan terlihat device baru yang terbaca pada other devices->gadget serials v2.4
- Klik kanan pada “Gadget Serial v2.4” lalu pilih “Update Driver Software”.



- Click “Browse my computer for driver software”. Lalu arahkan ke folder “arduino-1.5.3/hardware/arduino/x86/tools” lalu klik next maka windows akan menginstall driver yang dibutuhkan untuk koneksi ke Intel Galileo, tunggu sampai selesai.

[Browse for driver software on your computer](#)

Search for driver software in this location:

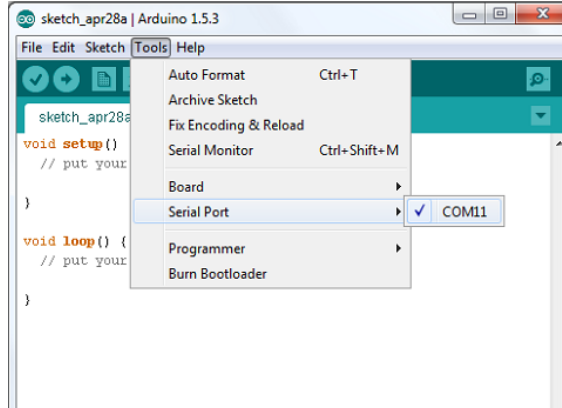
☒ Include subfolders

➔ Let me pick from a list of device drivers on my computer
This list will show installed driver software compatible with the device, and all driver software in the same category as the device.

- i. Setelah selesai, maka intel galileo sudah dapat digunakan dan tersambung pada COM di computer.



- j. Buka IDE. Pastikan Intel Galileo sudah terdeteksi oleh IDE.



Jika Serial Port sudah ada pada list, itu berarti penginstalan sudah berhasil dan siap untuk diprogram.

MODUL II

UPLOAD PROGRAM KE INTEL GALILEO

Untuk menulis kode yang akan diupload ke Intel Galileo, dibutuhkan Arduino Integrated Development Environment yang sudah terinstal pada computer atau biasa disingkat IDE, IDE merupakan suatu perangkat lunak yang digunakan untuk menulis kode program yang nantinya akan diupload pada Intel Galileo dan untuk mengecek apabila ada kesalahan atau error pada saat penulisan program. Hal ini karena Intel Galileo merupakan board yang sudah disertifikasi Arduino dan dapat digunakan sebagai pengganti Arduino Uno karena letak PIN in/out pada Intel Galileo sama dengan letak PIN yang ada pada Arduino Uno.

Tujuan

- Dapat menulis kode program pada IDE Arduino menggunakan bahasa C/C++
- Dapat mengupload kode program pada Intel Galileo

Dasar Teori

A. IDE

IDE (Integrated Development Environment) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak.

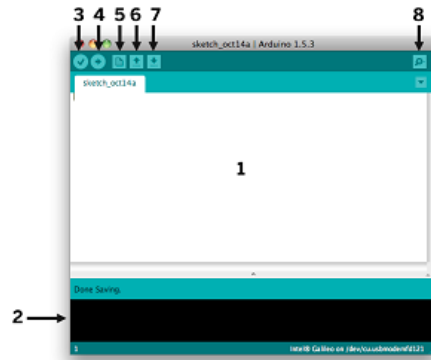
Sebuah IDE, atau secara bebas dapat diterjemahkan sebagai Lingkungan Pengembangan Terpadu, setidaknya memiliki fasilitas:

- **Editor**, yaitu fasilitas untuk menuliskan kode sumber dari perangkat lunak.
- **Compiler**, yaitu fasilitas untuk mengecek sintaks dari kode sumber kemudian mengubah dalam bentuk binari yang sesuai dengan bahasa mesin.
- **Linker**, yaitu fasilitas untuk menyatukan data binari yang beberapa kode sumber yang dihasilkan compiler sehingga data-data binari tersebut menjadi satu kesatuan dan menjadi suatu program komputer yang siap dieksekusi.
- **Debugger**, yaitu fasilitas untuk mengetes jalannya program, untuk mencari bug/kesalahan yang terdapat dalam program.

Sampai tahap tertentu IDE modern dapat membantu memberikan saran yang mempercepat penulisan. Pada saat penulisan kode, IDE juga dapat menunjukkan bagian-bagian yang jelas mengandung kesalahan atau keraguan.

B. Arduino IDE

Pada saat pertama kali membuka software Arduino IDE, kita akan melihat window untuk menulis program baru (sketch). Sketch adalah sebuah project Arduino yang berisi kode-kode program yang ditulis dalam bahasa C/C++. Berikut adalah keterangan dari bagian-bagian yang ada pada Arduino IDE

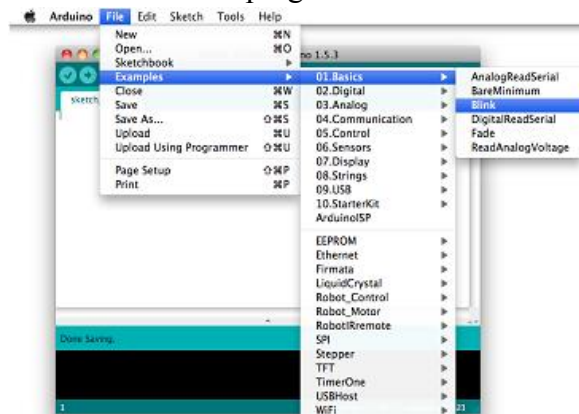


1. Code Entry Area
Area ini merupakan sebuah workspace atau tempat penulisan kode program.
2. Console
Merupakan sebuah pesan status yang digunakan untuk melihat keadaan apakah masih ada kesalahan pada penulisan kode program dan juga akan memuat status pada saat penguploadan kode program ke Intel Galileo
3. Verify Button
Klik tombol ini untuk memeriksa apakah kode program yang sudah dituliskan masih ada error. Jika ada error atau kesalahan penulisan, maka IDE akan memberitahukan programmer lewat Console
4. Upload Button
Tombol ini digunakan untuk mengupload kode program ke papan mikrokontroler apabila kode program sudah diverifikasi dan tidak ada error.
5. New Button
Digunakan untuk membuat Sketch baru.
6. Open Button
Tombol ini digunakan untuk membuka sketch atau kode program yang telah kita buat sebelumnya.
7. Save Button
Tombol ini digunakan untuk menyimpan sketch yang telah kita tulis dan dapat dibuka kembali pada waktu yang lain
8. Serial Monitor
Tombol ini akan membuka *serial monitor*, serial monitor adalah suatu interface yang dapat mengirim dan menerima informasi dari PC ke papan mikrokontroler.

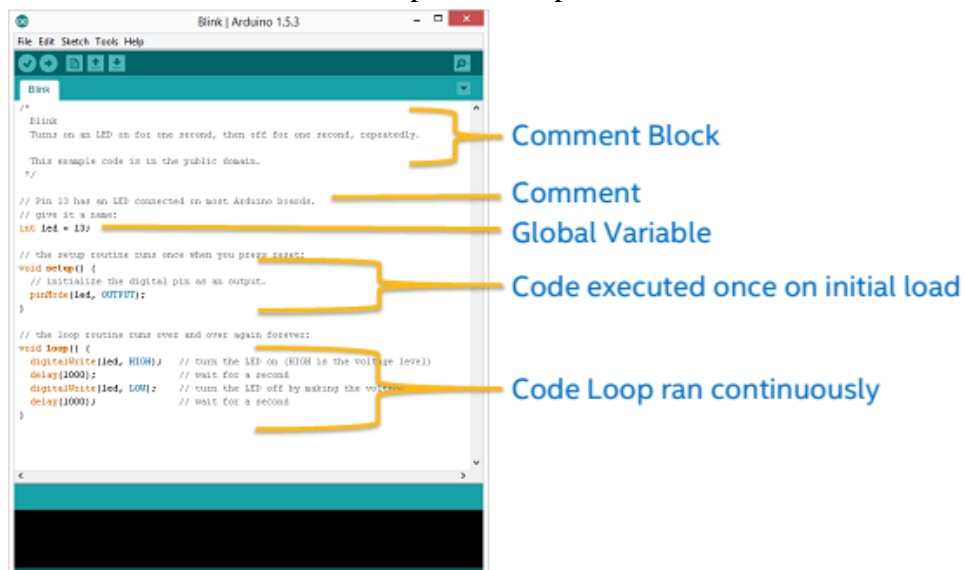
Langkah Praktikum

Pada praktikum ini diasumsikan praktikan sudah menginstall dan mengkonfigurasi Intel Galileo dan Arduino IDE pada computer dan sudah dapat digunakan.

1. Sambungkan kabel power terlebih dahulu pada intel galileo.
2. Sambungkan Intel Galileo dengan computer menggunakan kabel USB.
(Catatan : urutan pemasangan kabel harus selalu menyambungkan kabel power terlebih dahulu dan setelah itu penyambungan kabel USB, jika salah maka akan merusak papan mikrokontroler Intel Galileo.)
3. Buka Arduino IDE, lalu pastikan Intel Galileo terdeteksi pada Serial Port IDE.
4. Pada IDE, untuk meload contoh program klik File->Examples->01.Basics->Blink.
Maka pada workspace akan terisi oleh program contoh "Blink".



5. Setelah itu kode contoh akan terlihat pada workspace.



6. Setelah itu klik tombol upload pada tool bar.

7. Akan membutuhkan beberapa waktu untuk proses upload, kita akan melihat pesan status dari proses penguploadan pada console.
8. Setelah proses upload selesai, akan terlihat tulisan “Done uploading”.

Pada proses penguploadan, Arduino IDE sebenarnya sedang mengcompile program dari bahasa C/C++ menjadi intruksi-intruksi yang dapat dimengerti oleh prosesor Intel Galileo, setelah itu barulah Arduino IDE mengupload program yang sudah di-compile kedalam papan Galileo.

9. Setelah proses upload selesai, lihat pada papan board Galileo dan kita akan melihat lampu LED yang menyala dan redup setiap waktu yang terletak pada konektor batere. Jika tidak terjadi apa-apa, maka kita harus melakukan troubleshoot.



Troubleshoot

Berikut adalah langkah-langkah troubleshooting secara umum.

1. Pastikan Galileo tersambung dengan kabel power dan juga dengan computer via USB
2. Pastikan Intel Galileo terpilih pada Arduino IDE pada Tools->Board Menu
3. Gunakan Serial Port lain pada Tools->Serial Port
4. Kemungkinan firmware yang terinstall pada Galileo tidak update. Coba klik Help->Firmware Update pada saat Intel Galileo tersambung dan dalam keadaan menyala.
5. Pastikan kita memilih kode contoh “Blink” dan tidak mengubah kodenya.
6. Lihat pada console untuk mendapatkan petunjuk apabila terjadi error.

MODUL III

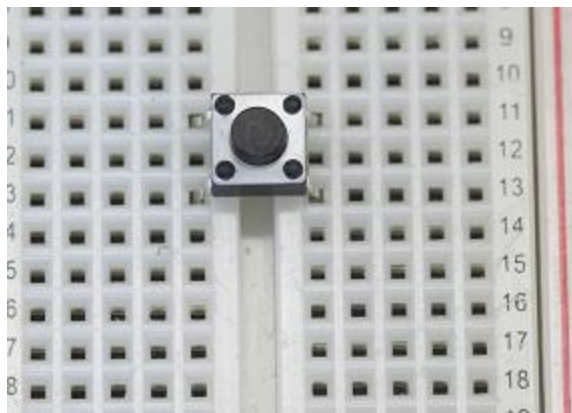
DIGITAL INPUT OUTPUT

Dalam penggunaan pin digital hanya akan mengenal logika “1” dan “0” atau istilah yang sering digunakan adalah “HIGH” dan “LOW”. Dimana fungsi yang digunakan pada IDE adalah `digitalRead()` untuk membaca suatu logika inputan pin dan `digitalWrite()` digunakan untuk menuliskan atau menentukan logika output.

Dasar Teori

A. Pushbutton

Pada praktikum ini Pushbutton merupakan sebuah komponen yang digunakan untuk menginputkan suatu logika ke pin Intel Galileo. Pushbutton dirangkai dengan sistem pull-up yang berarti ketika tidak ditekan maka logika input adalah “HIGH” sedangkan ketika ditekan logika input akan berubah menjadi “LOW”.



B. `digitalRead(pin)`

Hal yang paling penting dalam praktikum kali ini adalah sintaks `digitalRead(nomor-pin)` dan `digitalWrite(nomor-port)`. Kode ini merupakan sebuah fungsi yang dapat membaca nilai pada pin yang telah diset sebagai input dan melakukan eksekusi untuk menentukan output. Setelah nilai terbaca, maka nilai tersebut akan disimpan pada sebuah variabel, nilai 0 untuk LOW dan nilai 1 untuk HIGH yang akan digunakan sebagai pembanding untuk output.

Tujuan

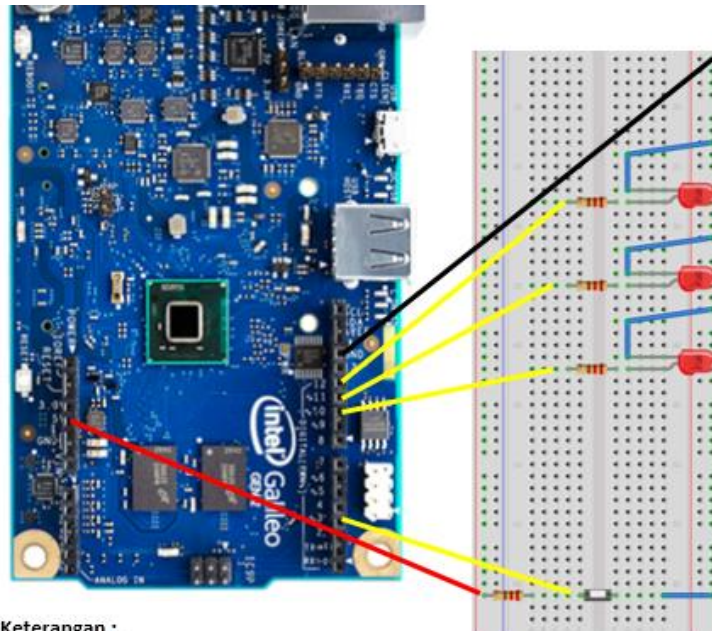
- Memahami Digital Input dan Output
- Dapat mengaplikasikan teori Digital Input dan Output

Alat yang dibutuhkan

- PC dan Intel Galileo
- Breadboard
- Kabel Jumper
- Pushbutton
- Resistor 10k ohm
- Resistor 220 Ohm
- LED 3x

Langkah Praktikum

1. Rangkai seperti Gambar dibawah :



Keterangan :

- Hitam : GND
- Merah : VCC
- Kuning : Sinyal I/O

2. Buatlah sketch baru pada Arduino IDE lalu ketikkan kode berikut

```
//Copy Dari Sini
#define LED1 10
#define LED2 11
#define LED3 12
int count = 0;
boolean Stat = true;

void setup()
{
  Serial.begin(9600);
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  digitalWrite(LED1, LOW);
  digitalWrite(LED2, LOW);
  digitalWrite(LED3, LOW);
  pinMode(3, INPUT);
}

void loop()
{
  Serial.println(count);
  if (digitalRead(3) == HIGH)
  {
    Stat = true;
  }
  else if (digitalRead(3) == LOW && Stat == true)
  {
    count++;
    if (count >= 6)
    {
      count = 0;
    }
    Stat = false;
  }
  if (count == 0)
  {
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    delay(100);
  }
  else if (count == 1)
  {
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
  }
  else if (count == 2)
  {
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
```

```

digitalWrite(LED3, HIGH);
delay(100);
digitalWrite(LED1, LOW);
digitalWrite(LED2, LOW);
digitalWrite(LED3, LOW);
delay(100);
}
else if (count == 3)
{
digitalWrite(LED1, HIGH);
delay(100);
digitalWrite(LED2, HIGH);
delay(100);
digitalWrite(LED3, HIGH);
delay(100);
digitalWrite(LED1, LOW);
delay(100);
digitalWrite(LED2, LOW);
delay(100);
digitalWrite(LED3, LOW);
delay(100);
}
else if (count == 4)
{
digitalWrite(LED3, HIGH);
delay(100);
digitalWrite(LED2, HIGH);
delay(100);
digitalWrite(LED1, HIGH);
delay(100);
digitalWrite(LED3, LOW);
delay(100);
digitalWrite(LED2, LOW);
delay(100);
digitalWrite(LED1, LOW);
delay(100);
}
else if (count == 5)
{
for (int i = 0; i <= 5; i++)
{
digitalWrite(LED1, HIGH);
delay(50);
digitalWrite(LED1, LOW);
delay(50);
digitalWrite(LED2, HIGH);
delay(50);
digitalWrite(LED2, LOW);
delay(50);
digitalWrite(LED3, HIGH);
delay(50);
digitalWrite(LED3, LOW);
delay(50);
digitalWrite(LED3, HIGH);
delay(50);
}
}
}

```

```

digitalWrite(LED3, LOW);
delay(50);
digitalWrite(LED2, HIGH);
delay(50);
digitalWrite(LED2, LOW);
delay(50);
digitalWrite(LED1, HIGH);
delay(50);
digitalWrite(LED1, LOW);
delay(50);
if (digitalRead(3) == LOW)
{
    count = 0;
    Stat = false;
    return;
}
}
delay(300);
for (int i = 0; i <= 7; i++)
{
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
    delay(30);
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
    delay(30);
    if (digitalRead(3) == LOW)
    {
        count = 0;
        Stat = false;
        return;
    }
}
delay(300);
}
}
//Sampai sini

```

3. Upload sketch tersebut ke Galileo
4. Buka serial lalu perhatikan yang terjadi apabila pushbutton ditekan dan perhatikan juga lima variasi LED.

MODUL IV

ANALOG INPUT OUTPUT PWM

Pada pembacaan analog akan berhubungan dengan resolusi bit ADC, dimana resolusi tersebut ditentukan oleh bit pengolahan. Nilai dari resolusi bit dapat diolah dan dapat dihubungkan dengan proses yang lain. PWM merupakan singkatan dari Pulse Width Modulation yang berarti representasi dari tegangan output ditentukan oleh dutty cycle dari logika “HIGH” dan “LOW” dalam satu periode.

Tujuan

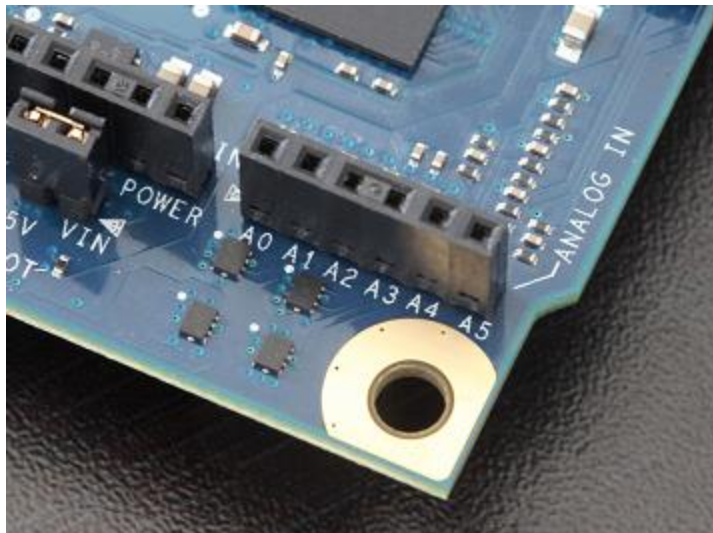
- Memahami Analog Input dan Output PWM
- Dapat mengaplikasikan teori Analog input dan Output PWM

Dasar Teori

A. Analog Digital Converter

Analog Digital Converter (ADC) adalah metode untuk mengubah suatu sinyal input analog ke nilai digital yang memiliki resolusi sesuai bit pengolahan. Pada modul ini akan digunakan ADC 10 bit yang berarti resolusi nilai yang akan kita peroleh adalah $2^{10}-1=1023$, resolusi dari tegangan 5 volt yang akan kita dapatkan adalah 0 s/d 1023.

Berikut gambar pin ADC yang dapat kita gunakan.



B. Pulse Width Modulation

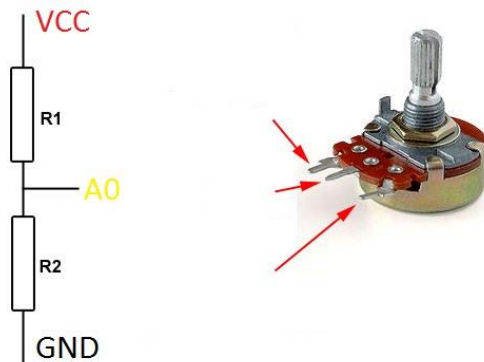
Pulse Width Modulation merupakan salah satu metode untuk merepresentasikan logika digital ke analog, dimana konsep yang digunakan adalah memanfaatkan ketidakmampuan sensitifitas komponen sisi output untuk mendeteksi pulsa yang sebenarnya, sehingga biasanya PWM relatif diatur dengan periode yang cepat.

Pada modul ini pwm yang digunakan adalah pwm 8 bit, berarti resolusi yang didapat adalah $2^{10}-1=255$. Karena input dari ADC yang kita dapat adalah 10 bit (1023) dan output yang kita gunakan adalah 8 bit (255), maka untuk pemrosesan harus diproses terlebih dahulu dengan cara :

$$\text{PWM output} = \text{Resolusi ADC} / \text{Resolusi PWM} = 1023 / 4$$

C. Potensio

Cara kerja potensi adalah mengubah nilai R1 dan R2 sesuai putaran dari sumbu pemutar. Perhatikan gambar berikut :



Nilai tegangan yang akan dibaca pada pin ADC adalah tegangan pada R2, dimana persamaanya adalah sebagai berikut :

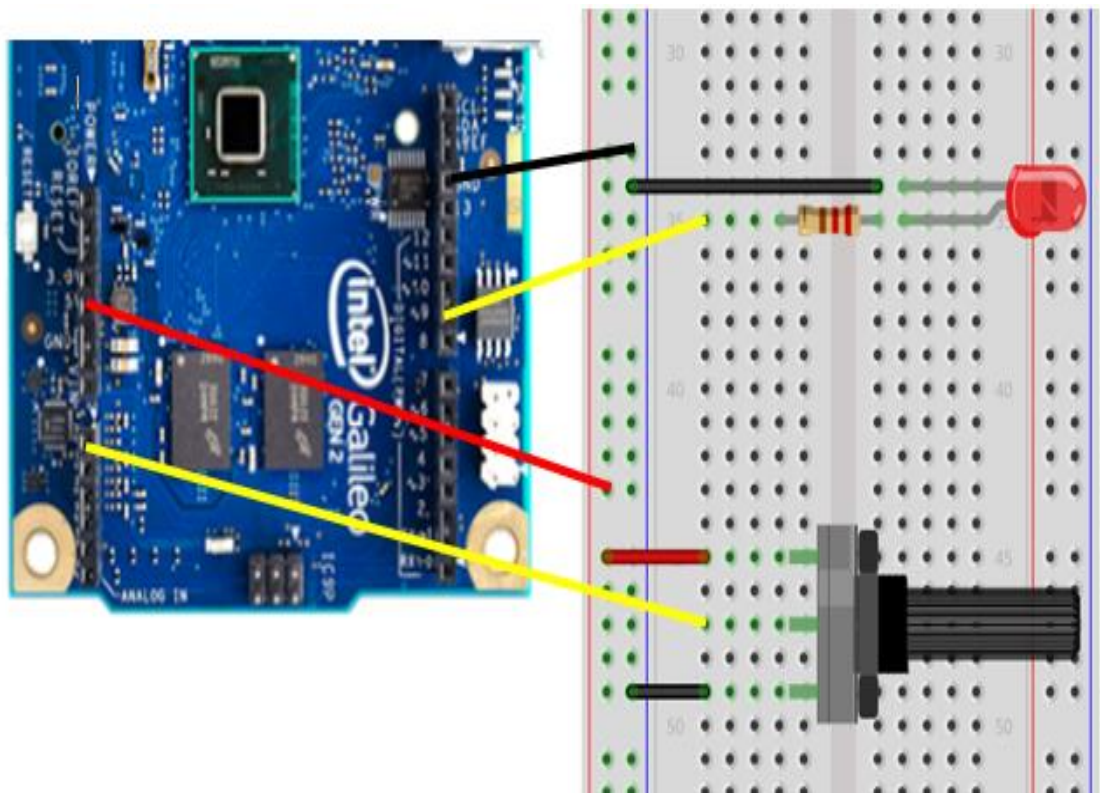
$$\begin{aligned} V_{R2} &= (R2 / R2 + R1) * V_{ref} \\ &= (R2 / R2 + R1) * 5 \\ &= \dots \text{ Volt (Variable berubah sesuai perubahan R1 dan R2)} \end{aligned}$$

Alat yang dibutuhkan

- PC dan Intel Galileo + Arduino IDE
- Breadboard
- Kabel Jumper
- Potensiometer 10k ohm
- Resistor 220 Ohm
- LED

Langkah Praktikum

1. Rangkai komponen seperti gambar berikut :



2. Buka Arduino IDE dan buat sketch baru, lalu tuliskan kode program dibawah ini :

```
#define ADC A0
#define LED 9
int val;
void setup()
{
  Serial.begin(9600);
  pinMode(LED,OUTPUT);
  analogWrite(LED,0);
}

void loop()
{
  val=analogRead(ADC)/4;
  analogWrite(LED,val);
  Serial.print("Nilai PWM = ");
  Serial.println(val);
}
```

3. Upload sketch tersebut ke Galileo.
4. Buka serial monitor.
5. Putar poros potensiometer lalu lihat perubahan nilai pwm di serial monitor dan perubahan kecerahan LED.

MODUL V

GPIO OVER LINUX

Dengan menggunakan command-line Interface pada linux, kita bisa mengontrol OS linux yang ada pada Intel galileo. Kita dapat menjalankan command untuk membaca atau menulis suatu file, mencari informasi tentang sistem yang tertanam, membuat koneksi jaringan, bahkan dapat membaca dan mengontrol pin yang terdapat pada Intel galileo. Pada modul ini kita akan mencoba untuk membaca input pin dan mengontrol output pin pada galileo menggunakan linux command-line interface.

Dasar Teori

A. GPIO

| Shield pin | Function | Linux | Interrupt L: H:high-level R:rising-edge F:falling-edge B:both edges | modes low-level |
|------------|----------|--------|--|--------------------|
| IO0 | UART0 RX | ttyS0 | - | |
| | GPIO | gpio11 | L/H/R/F | |
| IO1 | UART0 TX | ttyS0 | - | |
| | GPIO | gpio12 | L/H/R/F | |
| IO2 | UART1 RX | ttyS1 | | |
| | GPIO | gpio13 | L/H/R/F | |
| | GPIO | gpio61 | R/F/B | |
| IO3 | UART1 TX | ttyS1 | - | |
| | GPIO | gpio14 | L/H/R/F | |
| | PWM | pwm1 | - | |
| | GPIO | gpio62 | R/F/B | |
| IO4 | GPIO | gpio6 | R/F/B | |
| IO5 | GPIO | gpio0 | R/F/B | |
| | PWM | pwm3 | - | |
| IO6 | GPIO | gpio1 | R/F/B | |
| | PWM | pwm5 | - | |
| IO7 | GPIO | gpio38 | - | |
| IO8 | GPIO | gpio40 | - | |

| | | | |
|------|----------|-----------------|---------|
| IO9 | GPIO | gpio4 | R/F/B |
| | PWM | pwm7 | - |
| IO10 | GPIO | gpio10 | L/H/R/F |
| | PWM | pwm11 | - |
| IO11 | GPIO | gpio5 | R/F/B |
| | SPI MOSI | spidev1.0 | - |
| | PWM | pwm9 | - |
| IO12 | GPIO | gpio15 | L/H/R/F |
| | SPI MISO | spidev1.0 | - |
| IO13 | GPIO | gpio7 | R/F/B |
| | SPI SCK | spidev1.0 | - |
| IO14 | GPIO | gpio48 | R/F/B |
| | ADC A0 | in_voltage0_raw | - |
| IO15 | GPIO | gpio50 | R/F/B |
| | ADC A1 | in_voltage1_raw | - |
| IO16 | GPIO | gpio52 | R/F/B |
| | ADC A2 | in_voltage2_raw | - |
| IO17 | GPIO | gpio54 | R/F/B |
| | ADC A3 | in_voltage3_raw | - |
| IO18 | GPIO | gpio56 | R/F/B |
| | ADC A4 | in_voltage4_raw | - |
| | I2C SDA | i2c-0 | - |
| IO19 | GPIO | gpio58 | R/F/B |
| | ADC A4 | in_voltage5_raw | - |
| | I2C SCL | i2c-0 | - |

Langkah Praktikum

Pada modul ini diasumsikan praktikan sudah dapat mengkonfigurasi konektivitas dari Intel Galileo ke komputer dan dapat meremote intel galileo menggunakan telnet/SSH

1. Pastikan Intel Galileo dan komputer berada pada jaringan yang sama.
2. Lakukan telnet ke IP intel galileo
3. Login sebagai root dengan mengetikkan perintah “su”

```
/ # su
```

```
root@clanton:~#
```

4. Pindah ke direktori `/sys/class/gpio` :

```
root@clanton:~# cd /sys/class/gpio/
```

5. Sekarang list isi dari direktori tersebut menggunakan perintah `ls`:

```
root@clanton:/sys/class/gpio# ls
export      gpio19      gpio27      gpio38      gpio46      gpiochip0
gpio0       gpio20      gpio28      gpio4       gpio47      gpiochip16
gpio1       gpio21      gpio29      gpio40      gpio48      gpiochip2
gpio14      gpio22      gpio30      gpio41      gpio49      gpiochip8
gpio15      gpio23      gpio31      gpio42      gpio50      unexport
gpio16      gpio24      gpio32      gpio43      gpio51
gpio17      gpio25      gpio36      gpio44      gpio54
gpio18      gpio26      gpio37      gpio45      gpio55
```

List diatas merupakan daftar file dan folder yang digunakan untuk mengontrol suatu pin atau GPIO pada Intel Galileo (isi file pada linux yang berbeda atau pada Galileo yang berbeda mungkin tidak sama dengan gambar diatas). Dengan menggunakan Linux Command-line Interface, kita dapat menulis nilai pada file tertentu untuk mengontrol PIN sehingga PIN dapat dikontrol HIGH atau LOW. Selain itu, disini kita dapat membaca file tertentu untuk mendapatkan kondisi PIN saat ini.

Kata GPIO diatas memiliki kepanjangan General Purpose Input/Output, yang artinya PIN ini dapat di konfigurasi untuk melakukan banyak hal seperti digital output, digital input, analog output, dan analog input

6. Sambungkan LED ke PIN nomor 8. Karena PIN 8 tersambung pada Sinyal linux nomor 40, maka pindahlah ke direktori `gpio40` lalu list isi dari folder tersebut.

```
root@clanton:/sys/class/gpio# cd gpio40
```

```
root@clanton:/sys/class/gpio/gpio40# ls
```

```
active_low  direction  edge       power      subsystem  uevent
value
```

7. Command selanjutnya adalah “**cat**”, command ini digunakan untuk dapat melihat isi nilai dari file tertentu. Pertama-tama, lihat isi dari file **direction**.

```
root@clanton:/sys/class/gpio/gpio40# cat direction
```

```
in
```

Perintah diatas digunakan untuk melihat apakah pin dalam keadaan sebagai input atau sebagai output, jika keluaran dari “**cat direction**” adalah “**in**” maka pin tersebut terkonfigurasi sebagai input.

8. Untuk mengubahnya menjadi output, tulis “**out**” pada file **direction**. Yang artinya sama dengan perintah **pinMode(8,OUTPUT)** pada kode arduino.

```
root@clanton:/sys/class/gpio/gpio40# echo -n "out" > direction
```

9. Untuk mengatur pin 8 ke kondisi HIGH, cukup dengan menulis nilai 1 ke file **value**. Ini artinya sama dengan **digitalWrite(8,HIGH)** pada kode arduino.

```
root@clanton:/sys/class/gpio/gpio40# echo -n "1" > value
```

Jika kita melakukannya dengan benar, maka kita akan melihat LED yang dipasang pada PIN 8 menyala.

10. Untuk mematikan LED pada PIN 8 tadi, gunakan perintah:

```
root@clanton:/sys/class/gpio/gpio40# echo -n "0" > value
```

Jika kita menyalakan PIN 13 melalui kode arduino, LED yang terpasang on-board pada intel galileo juga akan ikut menyala, tetapi jika kita menggunakan linux GPIO untuk mengontrol PIN 13, hal ini tidak akan menyalakan on-board LED pada Galileo. Alasannya adalah karena PIN 13 dan on-board LED pada galileo hanya tersambung melalui software Arduino. Sebenarnya PIN 13 dan on-board LED pada Galileo memiliki sinyal GPIO yang berbeda pada linux.

Karena kita dapat membaca dan mengontrol PIN pada Galileo dengan cara menulis dan membaca file dengan command linux, hal ini membuka banyak kemudahan dan kemungkinan untuk memrogram Intel Galileo menggunakan bahasa selain arduino, contohnya seperti nodejs atau python.