



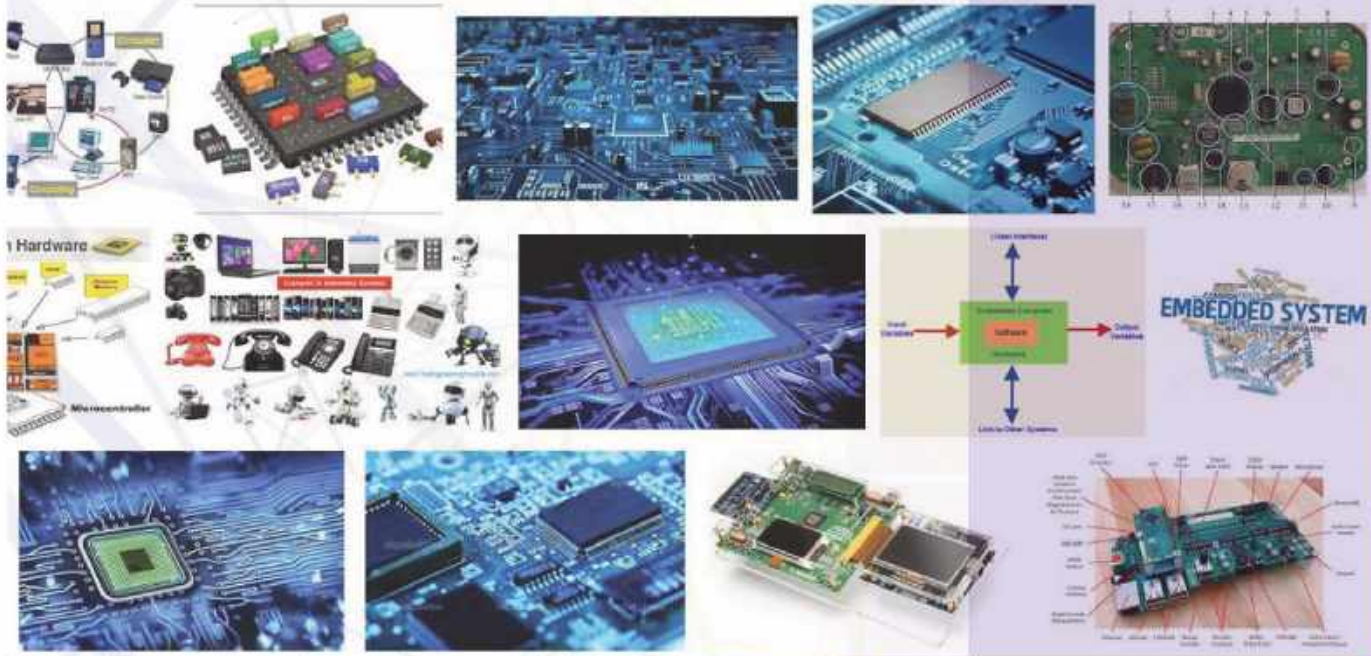
Embedded Systems



TE-SK

Modul Praktikum

EMBEDDED SYSTEM



Fakultas Teknologi Informasi dan Elektro
 Universitas Teknologi Yogyakarta
 2019

NIM : _____

Nama : _____

MODUL PRAKTIKUM SISTEM EMBEDDED

Oleh:
MS HENDRIYAWAN A

**UNIVERSITAS TEKNOLOGI YOGYAKARTA
2019**

@ 2019

Diterbitkan oleh :
Universitas Teknologi Yogyakarta
Jl. Siliwangi, Jombor, Sleman, Yogyakarta
Email : publikasi@uty.ac.id
Website : uty.ac.id

Modul Praktikum Sistem Embedded

ISBN
978-623-92626-9-3

Oleh : MS Hendriyawan A., S.T., M.Eng., Ph.D.

-

Edisi ke-1
Cetakan Pertama, 2019

Hak Cipta @2019 pada penulis,
Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku dalam bentuk apapun
tanpa ijin dari penulis.

KATA PENGANTAR

Teknologi elektronika maju dengan sangat pesat mengikuti kurva eksponensial dari tahun ketahun. Penerapannya hampir pada semua bidang kehidupan manusia modern, antara lain teknik, industri, transportasi, rumah tangga, lingkungan sosial, dll. Era modern saat ini sudah mulai meninggalkan teknologi elektronika digital tradisional dan berpindah menuju teknologi digital terprogram, yaitu sistem elektronika yang menerapkan kombinasi antara perangkat keras dan perangkat lunak terprogram, atau dikenal dengan sistem benam (embedded system). Sistem benam merupakan imbas dari ditemukannya teknologi pemroses mikro (microprocessor) yang memiliki kemampuan pengolahan aritmatika (arithmetic logic unit – ALU) dan pengendali proses (control unit – CU). Sistem benam merupakan komputer super mini yang mengintegrasikan fasilitas-fasilitas komputer skala besar dalam satu keping rangkaian terintegrasi (integrated circuit – IC), antara lain: microprocessor, memory, input-output, dll. Dengan adanya fasilitas tersebut dan dengan kesederhanaan implementasinya, membuat permintaan terhadap aplikasi sistem benam sangat besar oleh pengguna. Sehingga dengan permintaan yang besar dan bisa diproduksi secara massal maka harga jual sistem benam juga relatif murah, seperti contoh sistem benam berbasis Arduino. Terlebih saat ini dunia sudah mulai menerapkan isu Industrial 4.0 yaitu konsep penerapan teknologi IoT – Internet of Thing berbasis sistem benam untuk proses industri. Dengan fakta tersebut di atas, maka pembelajaran secara akademik tentang perancangan dan implementasi sistem benam sangat bermanfaat bagi mahasiswa perguruan tinggi. Upaya pembelajaran sistem benam akan memberikan nilai tambahan pengetahuan bagi mahasiswa untuk mempersiapkan diri bekerja di Industri modern yang sudah menerapkan konsep Industrial 4.0 untuk proses produksi.

Modul praktikum ini disusun oleh tim penulis yaitu M.S. Hendriyawan A., S.T., M.Eng. dengan menyajikan pembahasan terdiri ulasan teori dan praktik meliputi: pengenalan dasar sistem benam 8-bit, antarmuka masukan analog & digital, antarmuka keluaran analog & digital, komunikasi serial sinkron (UART) & tak sinkron (I2C, SPI), Real Time Operating System (RTOS), dan proyek yang merangkum seluruh materi.

Sebagai penutup, kiranya pasti ada kekurangan pada modul praktikum sistem benam ini baik dari segi tata tulis gaya bahasa maupun kedalaman materi yang disampaikan. Untuk itu kami akan senantiasa memperbaiki kualitas isi penyampaian materi agar selalu sinkron dengan target capaian pembelajaran dari waktu ke waktu. Saran dan koreksi bisa dikirimkan melalui e-mail penulis utama: hendriyawanachmad@uty.ac.id / hendriyawan.achmad@staff.uty.ac.id

Yogyakarta, 2019

Penulis

Daftar Isi

Halaman Judul		i
Kata Pengantar		iv
Daftar isi		v
Bab-I	Pengenalan Sistem Benam Berbasis Pengendali Mikro 8-bit	1
Bab-II	Antarmuka Masukan Analog	8
Bab-III	Antarmuka Masukan Digital	14
Bab-IV	Antarmuka Keluaran Analog	19
Bab-V	Antarmuka Keluaran Digital	25
Bab-VI	Komunikasi Digital UART	30
Bab-VII	Komunikasi Serial Sinkron – I2C	35
Bab-VIII	Komunikasi Serial Sinkron – SPI	41
Bab-IX	Real Time Operating System (RTOS)	48
Bab-X	Proyek-1: Mengukur dan Menampilkan Suhu	54
Bab-XI	Proyek-2: Data Logger Suhu	57
Bab-XII	Proyek-3: Pengendali Suhu Otomatis	60

I. PENGENALAN SISTEM BENAM BERBASIS PENGENDALI MIKRO 8-BIT

TUJUAN PEMBELAJARAN

Setelah melakukan praktik pada sesi ini, diharapkan mahasiswa mengetahui dan memahami beberapa hal berikut:

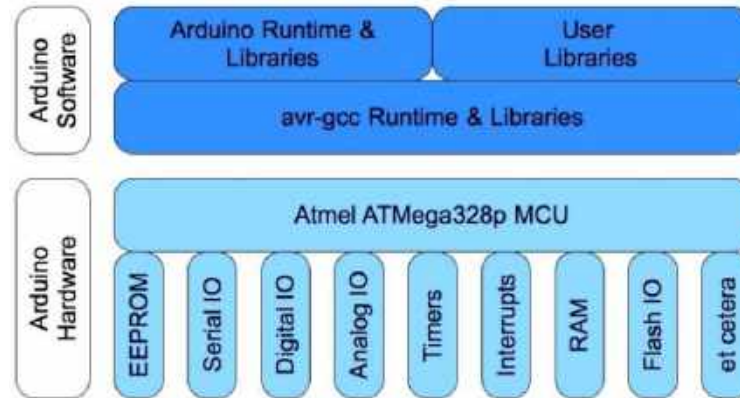
1. Pengertian tentang sistem benam.
2. Metode pengembangan sistem benam berbasis MCU 8-bit keluarga Arduino, meliputi perangkat keras dan perangkat lunak.

INDIKATOR CAPAIAN PEMBELAJARAN

1. Mahasiswa mampu menjelaskan definisi sistem benam berdasarkan arsitektur teknologinya.
2. Mahasiswa mampu mengoperasikan sistem pengembang MCU 8-bit Arduino dengan baik, meliputi perangkat lunak maupun perangkat keras.

TEORI SINGKAT

Embedded system atau dikenal dalam terjemahan Bahasa Indonesia sebagai sistem benam adalah sistem elektronik pengendali proses berbasis teknologi *Microcontroller*. Terdapat banyak jenis pengendali mikro yang umum digunakan ditinjau baik dari pembuat (misal: Atmel, Intel, Microchip, Motorola, dll.) maupun arsitekturnya (misal: 8-bit, 16-bit, atau 32-bit). Perbedaan antara pengendali mikro (*Microcontroller*) dan pemroses mikro (*Microprocessor*) adalah berdasarkan arsitektur internalnya. Pada pemroses mikro hanya menjalankan fungsi sebagai ALU (*Arithmetic Control Unit*) dan CU (*Control Unit*) saja, dan memerlukan piranti lain seperti memori dan I/O (Input/Output) agar dapat bekerja sebagai komputer. Sedangkan pengendali mikro memiliki arsitektur internal yang lengkap layaknya sebuah komputer dalam ukuran kecil dimana sudah terdapat piranti pemroses mikro, piranti memori (RAM – *random access memory*, ROM – *read only memory*), piranti IO, piranti Timer/Counter, piranti ADC/DAC, dsb. Teknologi saat ini memberikan kontribusi bagi produksi pengendali mikro sehingga memiliki fasilitas yang semakin lengkap sesuai dengan kebutuhan para pemakai yang semakin bervariasi. Gambar 1.1 menunjukkan arsitektur dasar dari sistem benam jenis 8-bit berbasis Arduino seri MCU ATmega328p yang diproduksi oleh Atmel.



Gbr. 1.1 Arsitektur perangkat lunak dan perangkat keras sistem benam Arduino 8-bit

Modul praktikum sistem benam yang akan digunakan sebagai rujukan aktifitas percobaan ini menggunakan sistem pengembang minimum dari keluarga Arduino™ yang berbasis pengendali mikro jenis 8-bit produksi Atmel. Sistem benam berbasis keluarga Arduino memiliki banyak variasi, antara lain: Arduino DUEMELANOVE, UNO, MEGA, LEONARDO, NANO, dll. Setiap varian memiliki fasilitas baik perangkat lunak maupun perangkat keras yang berbeda-beda. Alasan utama penggunaan sistem benam dari keluarga Arduino adalah: harganya murah, *cross platform* (Windows, Ubuntu, atau Macintosh), pemrogramannya mudah, *open source*, dan perangkat lunak/keras nya mudah dikembangkan.

ALAT DAN BAHAN PERCOBAAN

Pada sesi ini akan diperkenalkan tentang sistem pengembang MCU 8-bit Arduino yang meliputi perangkat keras dan perangkat lunak.

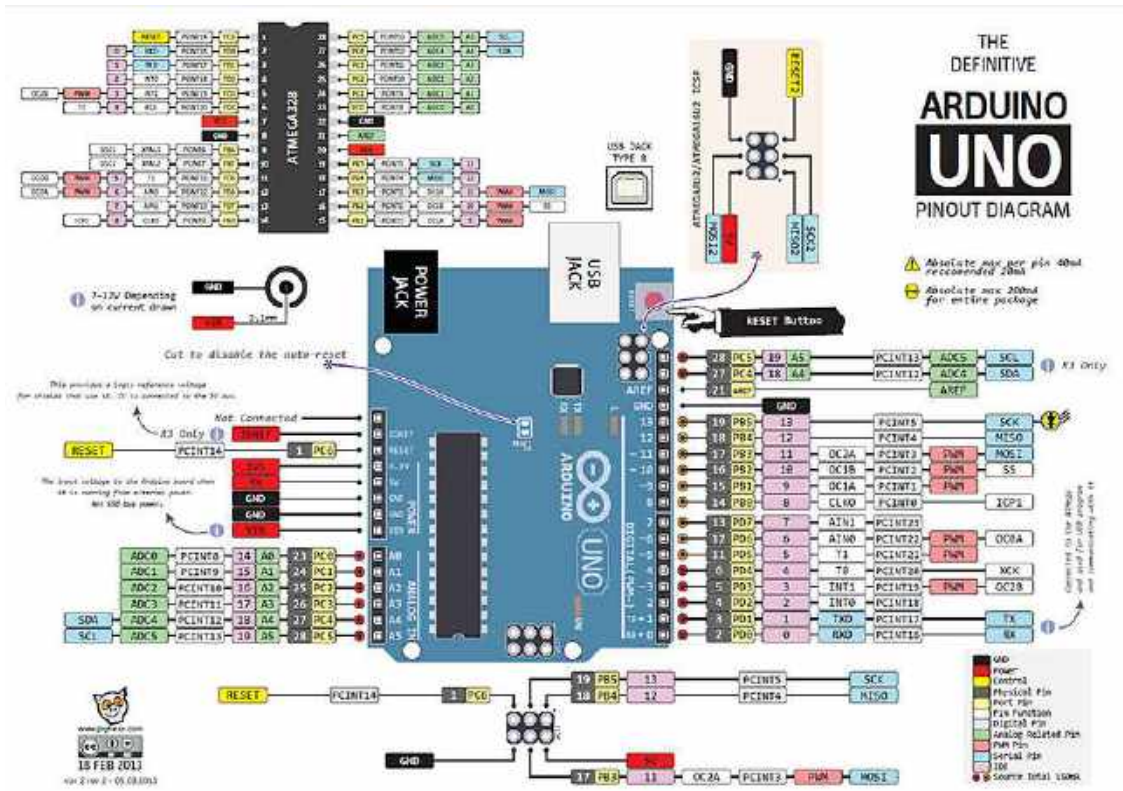
1. Arduino Board

Adalah perangkat keras elektronik yang terdapat sistem benam di dalamnya. Terdapat beberapa varian papan elektronik Arduino yang umum digunakan seperti ditunjukkan oleh Gambar 1.2 berikut ini.



Gbr. 1.2 Variasi papan elektronik Arduino. (Nano – Uno – Mega)

Pada umumnya sistem benam berbasis Arduino menggunakan jenis pengendali mikro ATMEL AVR 8-bit, antara lain: ATmega8, ATmega32, ATmega168, ATmega328, ATmega1280, dan ATmega2560. Selain itu, Arduino juga menggunakan jenis pengendali mikro 32-bit dari beberapa vendor, yaitu: SAM3X8E ARM Cortex-M3 produksi ATMEL, dan Galileo serta Curie produksi INTEL. Perkembangan perangkat keras Arduino sangatlah dinamis mengikuti keinginan pengguna yang menghendaki sebuah piranti elektronis serbaguna, mudah diprogram, harganya murah, dan memiliki fleksibilitas integrasi dengan piranti lain.



Gbr. 1.3 Tata letak dan penamaan pin masukan dan keluaran Arduino UNO

Tata letak dan penamaan pin masukan dan keluaran papan Arduino memiliki standar tersendiri, seperti yang ditunjukkan oleh Gambar 1.3 yang menunjukkan tata letak dan penamaan papan Arduino UNO. Pada dasarnya penamaan pin Arduino berdasarkan karakteristiknya, yaitu: input/output digital, dan input/output analog. Pin input/output A0 – A7 secara khusus digunakan untuk membaca data analog, namun secara umum juga bisa digunakan untuk input/output data digital. Selain itu terdapat pin 0, 1, 2, ... , n yang secara umum digunakan untuk input/output data digital. Maksimal nilai penomoran pin digital sesuai dengan jenis pengendali mikronya. Untuk papan Arduino UNO, pin input/output digital antara 0 – 13. Pada beberapa pin digital terdapat penambahan tanda ~ di awal nomor, hal ini menunjukkan bahwa pin juga bisa digunakan

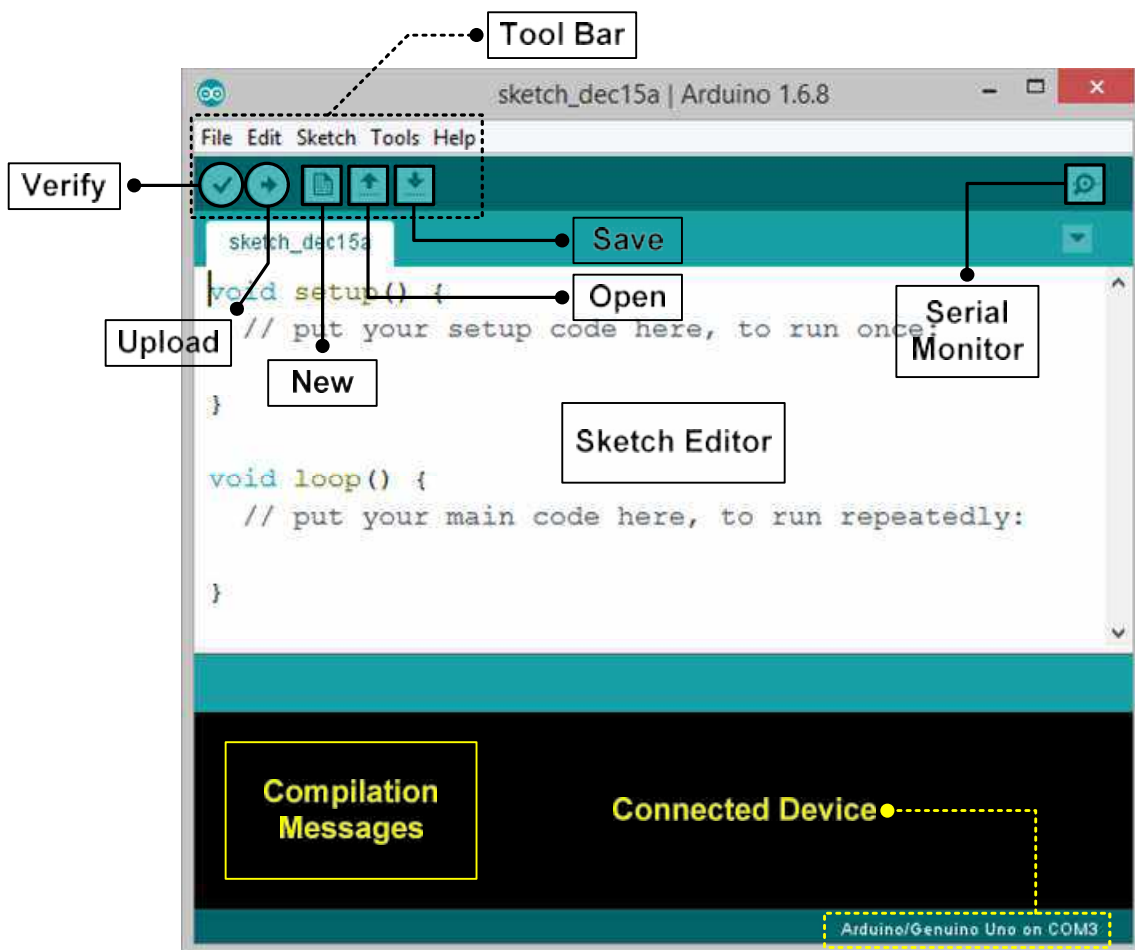
untuk output analog (PWM – *pulse width modulation*). Adapun fungsi-fungsi khusus lain bisa dilihat secara jelas pada Gambar 1.3.

2. Arduino IDE

Sistem Arduino tidak hanya sebatas perangkat keras elektronik saja, namun juga terdapat perangkat lunak pengembang nya, dikenal dengan Arduino IDE (*Integrated Development Environment*). Arduino IDE merupakan perangkat lunak pengembang berbasis platform Java yang digunakan untuk melakukan beberapa hal utama, antara lain:

- Menulis/mengubah kode program Arduino (sketch)
- Melakukan kompilasi kode program Arduino menjadi kode mesin (*.hex)
- Mengisikan kode mesin hasil kompilasi ke dalam pengendali mikro Arduino

Selain melakukan tiga hal utama di atas, Arduino IDE juga memiliki fasilitas pendukung lain, antara lain: pembaharuan library secara daring yang akan memudahkan dalam penyusunan kode program, monitor komunikasi data serial, dan pengaturan koneksi dengan perangkat keras Arduino. Gambar 1.4 menunjukkan tampilan Arduino IDE.



Gbr. 1.4 Tampilan Arduino IDE dengan keterangan fungsi tiap bagiannya.

Kode-kode program yang ditulis pada editor teks Arduino IDE disebut dengan **Sketch**, dengan menggunakan platform dasar bahasa pemrograman C yang sudah disesuaikan untuk keperluan pengembangan sistem benam berbasis Arduino. Status dan informasi proses kompilasi dapat dipantau pada bagian bagian bawan Arduino IDE, meliputi kesalahan yang terdeteksi, tahapan kompilasi, dan hasil kompilasi.

```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Struktur dasar pemrograman Arduino secara umum terbagi menjadi dua bagian, yaitu **setup** dan **loop**. *setup* digunakan untuk mendefinisikan fungsi dan karakteristik fasilitas yang akan digunakan dan hanya dieksekusi sekali saja, sedangkan *loop* digunakan untuk menyatakan sekuensial kerja yang akan dilakukan secara berulang terus menerus. Di luar *setup* dan *loop* dapat digunakan untuk deklarasi variabel dan fungsi, deklarasi library, definisi pin input/output, dll. Di bawah ini contoh struktur dasar penyusunan kode program pada Arduino IDE.

LANGKAH PERCOBAAN

Sebagai pengenalan sistem Arduino baik perangkat keras maupun perangkat lunak, dilakukan uji coba pengoperasian fasilitas kedua piranti tersebut. Adapun langkah-langkah percobaannya yaitu:

1. Menghubungkan papan Arduino dengan PC/Laptop melalui kabel USB.

Pada tahap ini papan Arduino tidak perlu diberikan daya listrik melalui konektor *power supply* karena saat terhubung dengan PC/Laptop melalui kabel USB maka papan Arduino secara langsung mendapatkan daya listrik. Kecuali jika papan Arduino terhubung dengan beban listrik yang besar (misal: motor servo, dll) maka harus diberikan daya listrik eksternal.

2. Membuka aplikasi Arduino IDE.

Setelah aplikasi Arduino IDE berhasil dijalankan, selanjutnya membuka contoh kode program (Blink.ino) yang sudah disediakan oleh Arduino IDE. Langkahnya:

Sorot: **File** → **Examples** → **01.Basics** → **Blink**

3. Mengatur koneksi Arduino IDE dengan papan Arduino

Hal pertama yang harus dilakukan adalah memilih jenis papan Arduino yang terhubung dengan PC/Laptop. Langkahnya:

Sorot: **Tools** → **Board:** “ ? ” dan pilih jenis papan Arduino yang digunakan.

Tahap selanjutnya adalah memilih port USB PC/Laptop yang sudah terhubung dengan papan Arduino. Langkahnya:

Sorot: **Tools** → **Port:** “ ? ” dan pilih nama port pada PC/Laptop.

4. Melakukan kompilasi kode program dan upload kode mesin menuju papan Arduino.

Setelah contoh kode program tertampil pada Sketch Editor, selanjutnya lakukan kompilasi kode program dengan menekan **ikon kompilasi** seperti yang terlihat pada Gambar 1.4. Jika tidak ada notifikasi kesalahan selanjutnya dilakukan proses upload kode mesin hasil kompilasi menuju papan Arduino yang sudah dihubungkan ke PC/Laptop dengan menekan **ikon upload** seperti yang terlihat pada Gambar 1.4.

5. Mengamati apa yang terjadi baik pada papan Arduino maupun pada Arduino IDE.

TUGAS

Berikut ini deskripsi tugas yang harus anda lakukan terkait dengan tahap-tahap percobaan yang sudah dilakukan.

1. Buka contoh kode program (selain Blink.ino) yang disediakan oleh Arduino IDE.
2. Lakukan pengaturan koneksi dengan papan Arduino.
3. Kompilasi dan upload kode program yang sudah anda buat, kemudian amati hasilnya.
4. Lakukan upload kode mesin pada papan Arduino.

this page is intentionally left blank

II. ANTARMUKA MASUKAN ANALOG

TUJUAN PEMBELAJARAN

Setelah melakukan praktik pada sesi ini, diharapkan mahasiswa mengetahui dan memahami beberapa hal berikut:

1. Cara menggunakan fasilitas ADC (*analog to digital converter*) pada papan Arduino untuk membaca input analog.
2. Prinsip kerja konversi data analog ke digital oleh ADC 10-bit.

INDIKATOR CAPAIAN PEMBELAJARAN

1. Mahasiswa mampu menjelaskan cara menggunakan fasilitas ADC Arduino untuk mengubah input analog menjadi data digital dan mengerti prinsip kerjanya.
2. Mahasiswa mampu membuat aplikasi sederhana untuk membaca input analog dari sensor dan mengubahnya menjadi informasi digital dengan benar.

TEORI SINGKAT

Arduino memiliki salah satu fasilitas penting, yaitu ADC (*Analog to Digital Converter*). Fasilitas ini bertugas untuk mengubah besaran input analog menjadi digital, karena pada dasarnya pengendali mikro adalah piranti digital yang tidak bisa mengenali data yang bersifat analog. Besaran alam (contoh: suhu, tekanan udara, kelembaban, dll) dapat dibaca menggunakan sensor yang mengeluarkan arus atau tegangan secara analog, padahal pengendali mikro hanya bisa mengenali data digital. Untuk itulah fungsi ADC diperlukan, agar pengendali mikro bisa membaca besaran alamiah yang terbaca oleh sensor analog.

Fasilitas ADC yang digunakan pada papan Arduino 8-bit adalah ADC 10-bit sebanyak 6 kanal yang terdapat pada pin **A₀**, **A₁**, **A₂**, **A₃**, **A₄**, **A₅**. Selain pin masukan analog, pada papan Arduino juga terdapat pin masukan tegangan referensi untuk ADC yaitu pin **A_{REF}**. Secara default, tegangan referensi ADC Arduino adalah +5.0 volt, namun dengan menggunakan pin **A_{REF}** bisa diberikan masukan bervariasi. Kalkulasi konversi analog ke digital oleh ADC mengikuti kaidah yang ditunjukkan oleh Persamaan 2.1 berikut ini:

$$Digital_{OUT} = \frac{2^N \times V_{masukan_analog}}{V_{Referensi}} \quad (1)$$

Data digital keluaran memiliki satuan **bit**, sedangkan **N** menunjukkan jumlah bit ADC yang dimiliki oleh papan Arduino, yaitu $N=10$ untuk jenis Arduino dengan pengendali mikro 8-bit. V_{analog} adalah tegangan analog yang terima oleh ADC, sedangkan $V_{referensi}$ adalah tegangan referensi yang digunakan oleh papan Arduino pada pin V_{REF} (default = +5.0 volt).

ALAT DAN BAHAN PERCOBAAN

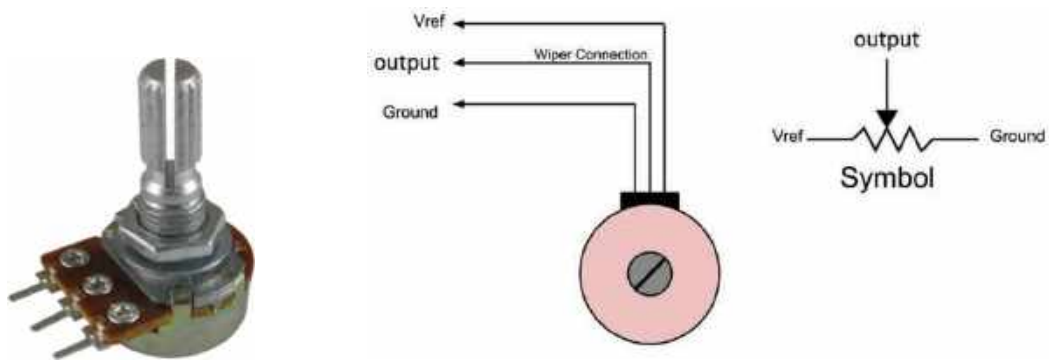
Pada sesi ini akan diperkenalkan tentang cara pembacaan masukan analog pada papan Arduino. Adapun alat dan bahan yang digunakan selama percobaan adalah:

1. Multimeter Digital

Alat bantu pengukur tagangan yang akan digunakan untuk mengukur tegangan masukan analognya sebagai data untuk keperluan analisis kerja ADC.

2. Potensiometer

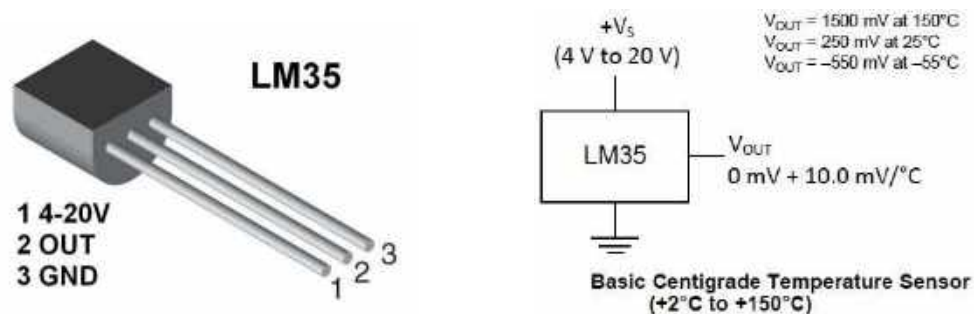
Bahan percobaan yang digunakan sebagai piranti masukan analog. Fungsi potensiometer adalah memberikan variasi masukan tegangan analog (0 – 5 volt) kepada papan Arduino berdasarkan gerakan putaran ke kiri dan ke kanan, seperti yang ditunjukkan oleh Gambar 2.1 berikut ini.



Gbr. 2.1 Potensiometer

3. Sensor Suhu LM35

Bahan percobaan yang digunakan sebagai sensor suhu dengan keluaran tegangan analog linier terhadap perubahan suhu. Fungsi dari piranti ini adalah memberikan informasi suhu yang akurat berupa tegangan analog mewakili nilai suhu yang terukur. Gambar 2.2 menunjukkan informasi teknis tentang IC LM35.



Gbr. 2.2 Tataletak kaki dan informasi teknis sensor suhu LM35

4. Papan Arduino UNO

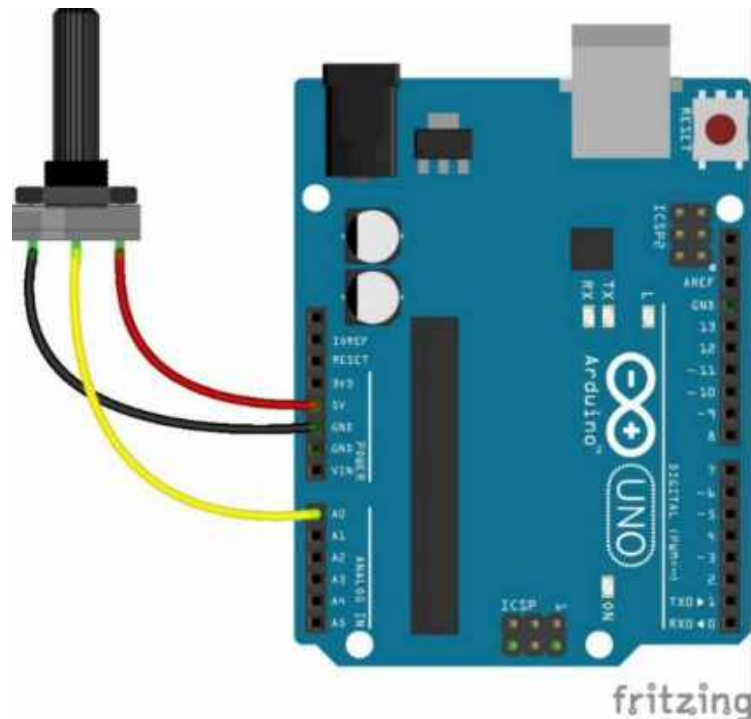
Bahan percobaan sebagai pengendali utama untuk menguji fungsi pembacaan masukan analog dengan melibatkan aktivitas perancangan antarmuka perangkat keras dan pemrograman perangkat lunak.

LANGKAH PERCOBAAN

Percobaan pembacaan masukan analog oleh papan Arduino melibatkan dua aktifitas, yaitu perancangan antarmuka perangkat keras dan pemrograman perangkat lunak. Adapun langkah percobaannya adalah sebagai berikut:

1. Disain antarmuka perangkat keras

Percobaan ini melibatkan aktifitas menghubungkan papan Arduino dengan potensiometer secara benar, seperti yang ditunjukkan oleh Gambar 2.3 berikut:



Gbr. 2.3 Antarmuka papan Arduino UNO dengan Potensiometer

2. Pemrograman perangkat lunak

Setelah perangkat keras selesai dirakit, selanjutnya membuat kode program untuk membaca masukan analog dari potensiometer. Ketik contoh kode program berikut ini, kompilasi dan upload ke papan Arduino. Kemudian pengamatan dilakukan menggunakan monitor data serial pada Arduino IDE.

```

#define potensiometer A0
int ADC_OUT;

void setup() {
  Serial.begin(9600);
  pinMode(potensiometer, INPUT);
}

void loop() {
  ADC_OUT = analogRead(potensiometer);
  Serial.print("Nilai digital = ");
  Serial.println(ADC_OUT);
  delay(1000);
}

```

3. Analisis hasil percobaan

Berdasarkan data pengukuran tegangan masukan analog dengan multi-meter digital dan berdasarkan pengamatan melalui monitor data serial, lakukan pencatatan hasilnya berdasarkan Tabel 2.1. Tahap selanjutnya adalah menganalisa kesalahan konversi ADC berdasarkan selisih antara hasil pengukuran multi-meter digital dengan kalkulasi secara teori matematis yang ditunjukkan oleh Persamaan 1.

Tabel 2.1 Perbandingan hasil pengamatan dengan teori fungsi ADC

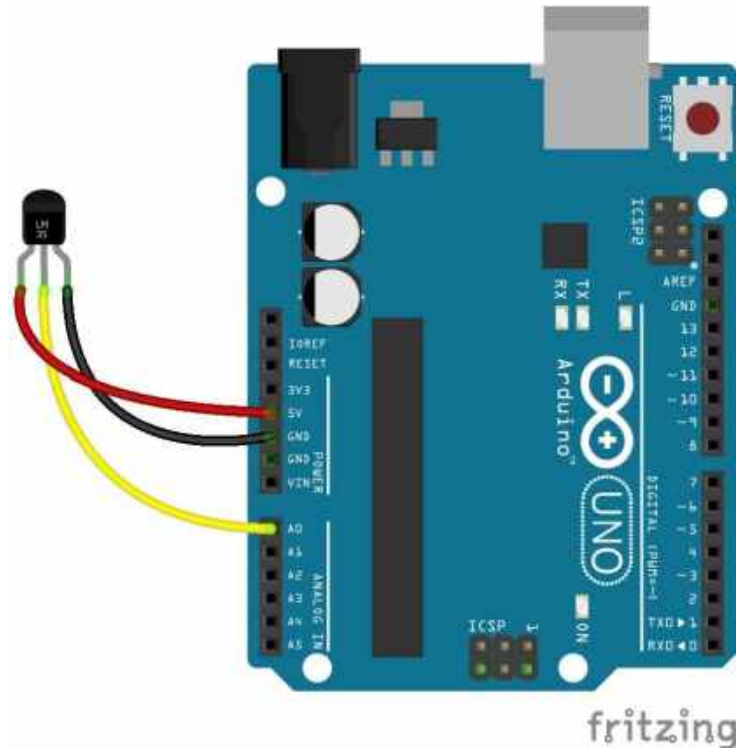
Sample	Pengukuran Masukan Analog	Kalkulasi Data Digital	Observasi Data Digital	Selisih
1	0.5 volts			
2	1.0 volts			
3	1.5 volts			
4	2.0 volts			
5	2.5 volts			
6	3.0 volts			
7	3.5 volts			
8	4.0 volts			
9	4.5 volts			

Selanjutnya hasil perbandingan dalam tabel 2.1 dibuat dalam bentuk grafik perbandingan menggunakan Microsoft Office Excel dan simpulkan.

TUGAS

Berikut ini deskripsi tugas yang harus anda lakukan terkait dengan tahap-tahap percobaan yang sudah dilakukan.

1. Merakit antarmuka perangkat keras antara sensor suhu LM35 dengan papan Arduino UNO seperti yang ditunjukkan Gambar 2.4 berikut ini.



Gbr. 2.4 Antarmuka papan Arduino UNO dengan LM35

2. Membuat kode program yang menerjemahkan masukan analog LM35 menjadi informasi suhu dalam satuan Centigrade ($^{\circ}\text{C}$) berdasarkan informasi teknis yang ditunjukkan oleh Gambar 2.2.
3. Kompilasi dan upload kode program yang sudah anda buat, kemudian amati hasilnya.
4. Membuat laporan percobaan yang berisi foto antarmuka perangkat keras dan kode program yang berhasil dibuat dan berjalan dengan benar.

this page is intentionally left blank

III. ANTARMUKA MASUKAN DIGITAL

TUJUAN PEMBELAJARAN

Setelah melakukan praktik pada sesi ini, diharapkan mahasiswa mengetahui dan memahami beberapa hal berikut:

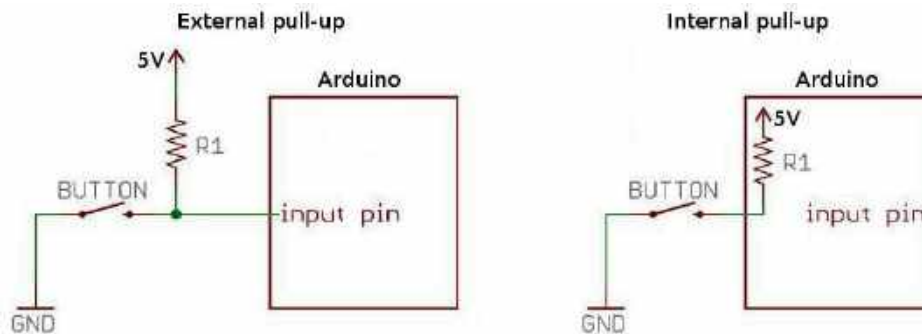
1. Cara menggunakan fasilitas GPIO (*general purpose input output*) pada papan Arduino untuk membaca input digital.
2. Prinsip kerja pin input digital dengan atau tanpa *pull up*.

INDIKATOR CAPAIAN PEMBELAJARAN

1. Mahasiswa mampu menjelaskan cara menggunakan fasilitas GPIO Arduino untuk membaca masukan digital dan mengerti prinsip kerjanya.
2. Mahasiswa mampu membuat aplikasi sederhana untuk membaca input digital dan menampilkan informasi masukan digital dengan benar.

TEORI SINGKAT

Arduino memiliki salah satu fasilitas penting selain ADC, yaitu GPIO (*General Purpose Input Output*). Fasilitas ini memiliki fungsi serbaguna yaitu bisa menerima input atau mengeluarkan output digital secara bergantian tergantung pada pengaturan fungsinya oleh kode program perangkat lunak. Berbeda dengan ADC yang menerima masukan analog, pin masukan digital Arduino hanya mampu mengenali data digital saja, yaitu kondisi **HIGH** (+5 volt) dan **LOW** (0 volt). Pada papan Arduino UNO, pin masukan digital tidak hanya pin 1 – 13 saja, namun pin masukan analog juga bisa digunakan sebagai pin masukan digital. Secara umum pin masukan digital Arduino memiliki resistor *pull_up* internal yaitu sebuah resistor yang menarik arus dari catu positif menuju catu negatif, namun secara default resistor ini tidak diaktifkan. Sehingga perlu dipertimbangkan penggunaannya berdasarkan karakteristik masukan dari piranti eksternal.



Gbr. 3.1 Penggunaan resistor pull up pada masukan digital Arduino

Jika piranti masukan digital yang terhubung dengan papan Arduino mampu mengalirkan arus listrik menuju pin masukan digital Arduino maka resistor *pull_up* tidak diperlukan, namun jika piranti masukan digital hanya bisa menyedot arus listrik dari pin masukan digital Arduino maka resistor *pull_up* internal harus digunakan agar terdapat beda potensial yang terbaca sebagai nilai HIGH atau LOW.

ALAT DAN BAHAN PERCOBAAN

Pada sesi ini akan diperkenalkan tentang cara pembacaan masukan analog pada papan Arduino. Adapun alat dan bahan yang digunakan selama percobaan adalah:

1. Saklar Tekan On – Off

Bahan percobaan yang digunakan untuk memberikan logika masukan digital.

2. Multimeter Digital

Alat bantu pengukur tagangan yang akan digunakan untuk mengukur tegangan masukan digitalnya sebagai data untuk keperluan analisis kerja GPIO.

3. Papan Arduino UNO

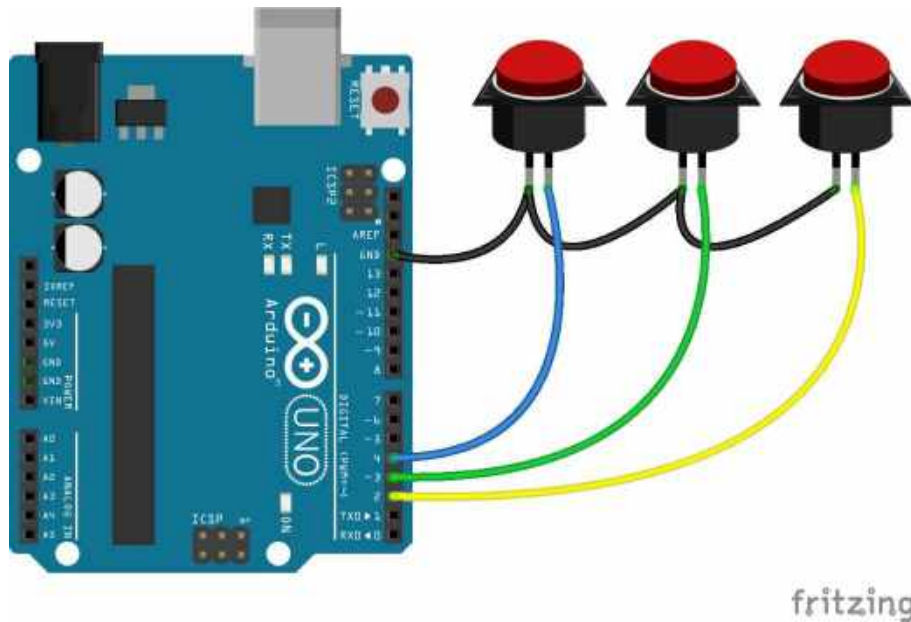
Bahan percobaan sebagai pengendali utama untuk menguji fungsi pembacaan masukan digital dengan melibatkan aktivitas perancangan antarmuka perangkat keras dan pemrograman perangkat lunak.

LANGKAH PERCOBAAN

Percobaan pembacaan masukan digital oleh papan Arduino melibatkan dua aktifitas, yaitu perancangan antarmuka perangkat keras dan pemrograman perangkat lunak. Adapun langkah percobaannya adalah sebagai berikut:

1. Disain antarmuka perangkat keras

Percobaan ini melibatkan aktifitas menghubungkan papan Arduino UNO dengan tiga saklar tekan on-off secara benar, seperti yang ditunjukkan oleh Gambar 3.2. Pada percobaan ini akan memanfaatkan resistor *pull up* internal yang dimiliki oleh Arduino UNO, sehingga tidak memerlukan tambahan resistor pull up eksternal.



Gbr. 3.2 Antarmuka papan Arduino UNO dengan Saklar Tekan On-Off.

2. Pemrograman perangkat lunak

Setelah perangkat keras selesai dirakit, selanjutnya membuat kode program untuk membaca masukan digital dari saklar tekan on-off. Ketik contoh kode program berikut ini, kompilasi dan upload ke papan Arduino. Kemudian pengamatan dilakukan menggunakan monitor data serial pada Arduino IDE.

```
#define sw1 2
#define sw2 3
#define sw3 4

void setup() {
  Serial.begin(9600);
  pinMode(sw1, INPUT_PULLUP);
  pinMode(sw2, INPUT_PULLUP);
  pinMode(sw3, INPUT_PULLUP);
}

void loop() {
  if(sw1 == 0) Serial.println("Tombol-1 ditekan");
  if(sw2 == 0) Serial.println("Tombol-2 ditekan");
  if(sw3 == 0) Serial.println("Tombol-3 ditekan");
  delay(100);
}
```

3. Analisis Hasil Percobaan

Analisa hasil percobaan dilakukan berdasarkan pengukuran tegangan menggunakan multimeter pada setiap pin masukan digital (2, 3, 4) baik saat kondisi ditekan maupun saat kondisi tidak ditekan. Hasil pengukuran disajikan dalam bentuk isian seperti yang ditunjukkan oleh Tabel 3.1 berikut ini.

Tabel 3.1 Hubungan logika masukan dengan tegangan yang terukur

Sample	Kondisi	V _{terukur} pada pin 2 (SW1)	V _{terukur} pada pin 3 (SW2)	V _{terukur} pada pin 4 (SW3)
1	<i>Ditekan</i>			
2	<i>Dilepas</i>			

Berdasarkan isian pada Tabel 3.1 di atas dapat disimpulkan hubungan antara logika masukan yang terdeteksi dengan tegangan yang terukur oleh multimeter. Silahkan tulis kesimpulan yang anda dapatkan sebagai pembuktian kebenaran logika digital masukan dilihat dari level tegangannya.

TUGAS

Berikut ini deskripsi tugas yang harus anda lakukan terkait dengan tahap-tahap percobaan yang sudah dilakukan.

1. Berdasarkan Gambar 3.2 silahkan anda buat kode program yang hanya menggunakan 2 saklar tekan on-off saja (silahkan pilih dari 3 saklar tekan yang ada). Aplikasi yang anda harus buat adalah fungsi COUNTER UP/DOWN, dimana salah satu saklar tekan sebagai penambah hitungan saat ditekan dan saklar tekan yang lain sebagai pengurang hitungan. Anda dapat menggunakan variabel bertanda maupun tak bertanda untuk menyimpan hitungan.
2. Kompilasi dan upload kode program yang sudah anda buat, kemudian amati hasilnya.
3. Membuat laporan percobaan yang berisi foto antarmuka perangkat keras dan kode program yang berhasil dibuat dan berjalan dengan benar.

this page is intentionally left blank

IV. ANTARMUKA KELUARAN ANALOG

TUJUAN PEMBELAJARAN

Setelah melakukan praktik pada sesi ini, diharapkan mahasiswa mengetahui dan memahami beberapa hal berikut:

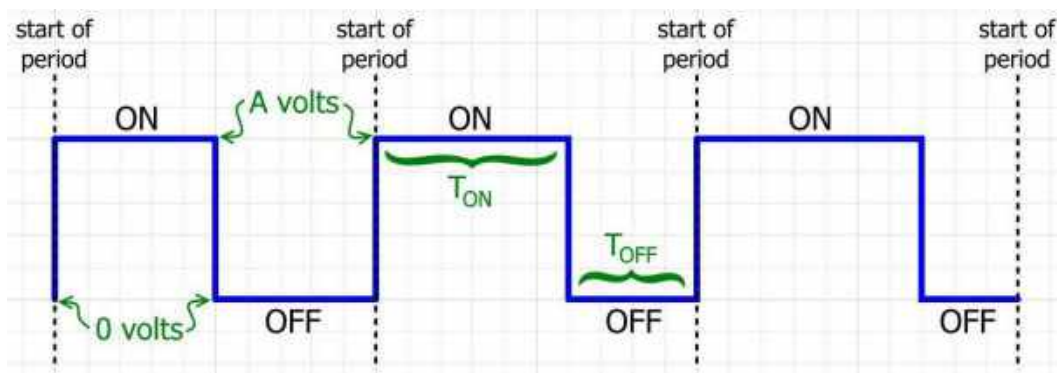
1. Prinsip kerja dari teknik PWM (*pulse width modulation*).
2. Cara menggunakan fasilitas keluaran analog PWM pada papan Arduino untuk mengubah data digital menjadi besaran keluaran analog.

INDIKATOR CAPAIAN PEMBELAJARAN

1. Mahasiswa mampu menjelaskan cara menggunakan fasilitas PWM Arduino untuk menghasilkan keluaran analog dan mengerti prinsip kerjanya.
2. Mahasiswa mampu membuat aplikasi sederhana tentang penerapan PWM untuk mengendalikan piranti keluaran secara analog.

TEORI SINGKAT

PWM atau lebih dikenal dengan modulasi lebar pulsa, yaitu sebuah metode pengaturan lebar pulsa secara variabel untuk sebuah nilai frekwensi yang tetap. Berbeda halnya dengan modulasi frekwensi (FM) atau modulasi amplitudo (AM) yang mengatur variasi frekwensi atau tegangan, PWM menggunakan gelombang kotak (*square wave*) dengan amplitudo dan frekwensi yang tetap namun rasio periode **ON** terhadap total periode gelombang atau disebut dengan **DUTY CYCLE** bisa diatur dalam satuan persen (%). Gambar 4.1 menunjukkan bentuk gelombang PWM.



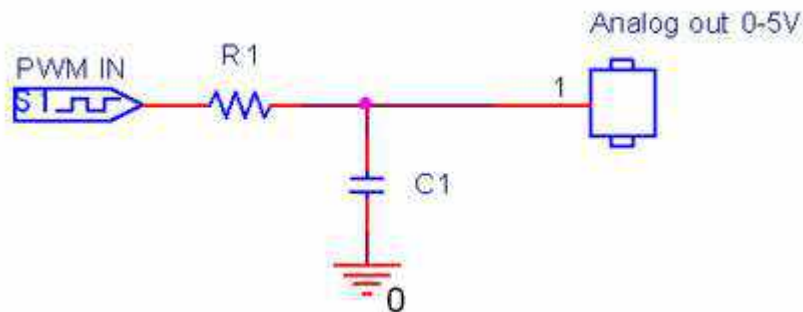
Gbr. 4.1 Bentuk gelombang PWM

Berdasarkan Gambar 4.1 yang menunjukkan bentuk gelombang kotak PWM, dapat dilihat bahwa periode ON pada satu siklus gelombang bisa melebar dan menyempit tanpa merubah frekwensi dan amplitudonya. Variasi lebar pulsa PWM dapat dihitung secara matematis mengikuti kaidah yang ditunjukkan oleh Persamaan 2 berikut.

$$duty\ cycle\ (\%) = \frac{T_{ON}}{T_{ON}+T_{OFF}} \quad (2)$$

Pada papan Arduino jenis 8-bit terdapat beberapa pin yang bisa dijadikan sebagai keluaran PWM, yaitu: 3, 5, 6, 9, 10, 11. Dan terdapat dua jenis frekwensi PWM yang berbeda, yaitu: **488 Hz** untuk pin **3, 9, 10, 11**, dan **977 Hz** untuk pin **5, 6**. Rentang amplitudo PWM Arduino hampir sama dengan tegangan catu komponen digital, yaitu: 0 – 5 volts. Sedangkan rentang nilai digital yang merepresentasikan *duty cycle* pada Arduino yaitu: 0 – 255 untuk 0% – 100% rasio.

Keluaran PWM Arduino belum dapat dikatakan sebagai keluaran analog murni karena pada dasarnya masih berupa variasi data digital (HIGH dan LOW), sehingga diperlukan sebuah tapis lolos rendah (LPF – *low pass filter*) untuk mendapatkan gradasi amplitudo mendekati definisi keluaran analog. Gambar 4.2 menunjukkan konfigurasi LPF untuk keluaran PWM.



Gbr. 4.2 LPF untuk keluaran PWM.

Dengan menggunakan LPF maka akan ada konversi dari duty cycle menjadi tegangan analog. Nilai R dan C pada LPF seperti ditunjukkan oleh Gambar 4.2 umumnya bernilai $R1 = 3.9K\Omega$ dan $C1 = 100nF$. Dengan demikian, nilai digital PWM Arduino yang diperlukan untuk menghasilkan keluaran analog dapat dihitung mengikuti kaidah yang ditunjukkan oleh Persamaan 3 berikut.

$$Digital_{PWM} = 255 \times \frac{Analog_{PWM}}{Amplitudo_{PWM}} \quad (3)$$

$Analog_{PWM}$ adalah tegangan analog yang diinginkan, dan $Amplitudo_{PWM}$ adalah amplitudo maksimal gelombang PWM ($\approx +5$ volts). Sehingga data digital 0 menghasilkan tegangan keluaran analog 0 volts (0% duty cycle), dan data digital 255 menghasilkan tegangan keluaran analog 5 volts (100% duty cycle) untuk Arduino yang menggunakan catu tegangan logika +5 volts.

ALAT DAN BAHAN PERCOBAAN

Pada sesi ini akan diperkenalkan tentang menghasilkan keluaran analog pada papan Arduino menggunakan fasilitas PWM. Adapun alat dan bahan yang digunakan selama percobaan adalah:

1. Resistor = $5.9K\Omega$ dan Capacitor = 100nF

Resistor dan kapasitor akan digunakan untuk membuat tapis lolos rendah (LPF) yang akan mengubah gelombang PWM menjadi tegangan analog.

2. Multimeter Digital

Alat bantu pengukur tagangan yang akan digunakan untuk mengukur tegangan analog keluaran LPF.

3. Papan Arduino UNO

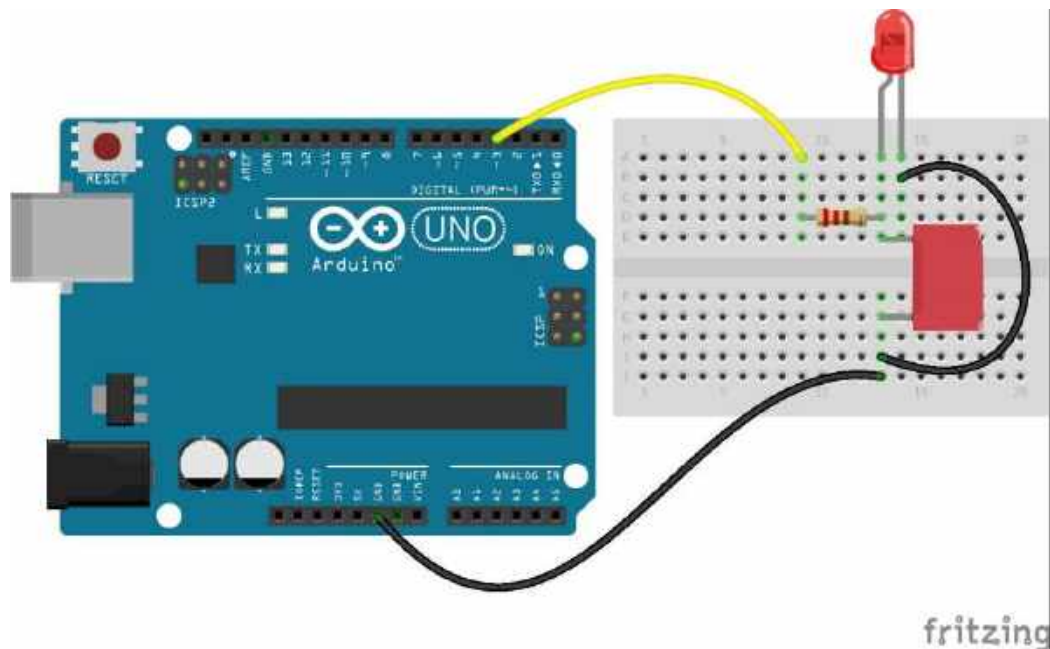
Bahan percobaan sebagai pengendali utama untuk menguji fungsi PWM dengan melibatkan aktivitas perancangan antarmuka perangkat keras dan pemrograman perangkat lunak.

LANGKAH PERCOBAAN

Percobaan fungsi keluaran analog dengan PWM oleh papan Arduino melibatkan dua aktifitas, yaitu perancangan antarmuka perangkat keras dan pemrograman perangkat lunak. Adapun langkah percobaannya adalah sebagai berikut:

1. Disain antarmuka perangkat keras

Percobaan ini melibatkan aktifitas menghubungkan pin keluaran PWM pada papan Arduino UNO dengan rangkaian LPF secara benar, seperti yang ditunjukkan oleh Gambar 4.2. Pada percobaan ini akan memanfaatkan resistor dan capacitor. Sebagai indikator tegangan keluaran analog digunakan LED. Gambar 4.3 menunjukkan antarmuka perangkat keras sebagai bahan percobaan.



Gbr. 4.3 Antarmuka papan Arduino UNO dengan rangkaian LPF.

2. Pemrograman perangkat lunak

Setelah perangkat keras selesai dirakit, selanjutnya membuat kode program untuk menggunakan akses keluaran analog PWM. Ketik contoh kode program berikut ini, kompilasi dan upload ke papan Arduino. Kemudian pengamatan dilakukan menggunakan monitor data serial pada Arduino IDE dan multimeter digital.

```
#define PWM 3 //definisi pin output PWM
void setup() {
  Serial.begin(9600); //inisialisasi serial
  pinMode(PWM, OUTPUT); //pin PWM sebagai output
}
void loop() {
  Serial.println("nilai PWM = 50");
  analogWrite(PWM, 50); //PWM = 50;
  delay(5000); //tunda pengamatan 5 detik
  Serial.println("nilai PWM = 100");
  analogWrite(PWM, 100); //PWM = 100;
  delay(5000); //tunda pengamatan 5 detik
  Serial.println("nilai PWM = 150");
  analogWrite(PWM, 150); //PWM = 150;
  delay(5000); //tunda pengamatan 5 detik
  Serial.println("nilai PWM = 200");
  analogWrite(PWM, 200); //PWM = 200;
  delay(5000); //tunda pengamatan 5 detik
}
```

3. Analisis Hasil Percobaan

Analisa hasil percobaan dilakukan berdasarkan pengukuran tegangan menggunakan multimeter pada pin keluaran analog LPF untuk setiap nilai digital PWM. Hasil pengukuran disajikan dalam bentuk isian seperti yang ditunjukkan oleh Tabel 4.1. Tahap selanjutnya adalah menganalisa kesalahan konversi nilai digital ke analog berdasarkan selisih antara hasil pengukuran multimeter dengan kalkulasi secara teori matematis yang ditunjukkan oleh Persamaan 3.

Tabel 4.1 Perbandingan hasil pengamatan dengan teori fungsi konversi PWM ke analog

Sample	Nilai Digital PWM	Kalkulasi Keluaran Analog (volts)	Observasi Keluaran Analog (volts)	Selisih
1	50			
2	100			
3	150			
4	200			

Selanjutnya hasil perbandingan dalam tabel 4.1 dibuat dalam bentuk grafik perbandingan menggunakan Microsoft Office Excel dan simpulkan.

TUGAS

Berikut ini deskripsi tugas yang harus anda lakukan terkait dengan tahap-tahap percobaan yang sudah dilakukan.

1. Membuat program COUNTER secara otomatis menggunakan fungsi delay dengan nilai cacahan antara 0 – 255, kemudian setiap nilai cacahan dijadikan nilai keluaran digital PWM. Amati tegangan keluaran LPF menggunakan multimeter, kemudian jelaskan hasil pengamatan anda.
2. Kompilasi dan upload kode program yang sudah anda buat, kemudian amati hasilnya.
3. Membuat laporan percobaan yang berisi foto antarmuka perangkat keras dan kode program yang berhasil dibuat dan berjalan dengan benar.

this page is intentionally left blank

V. ANTARMUKA KELUARAN DIGITAL

TUJUAN PEMBELAJARAN

Setelah melakukan praktik pada sesi ini, diharapkan mahasiswa mengetahui dan memahami beberapa hal berikut:

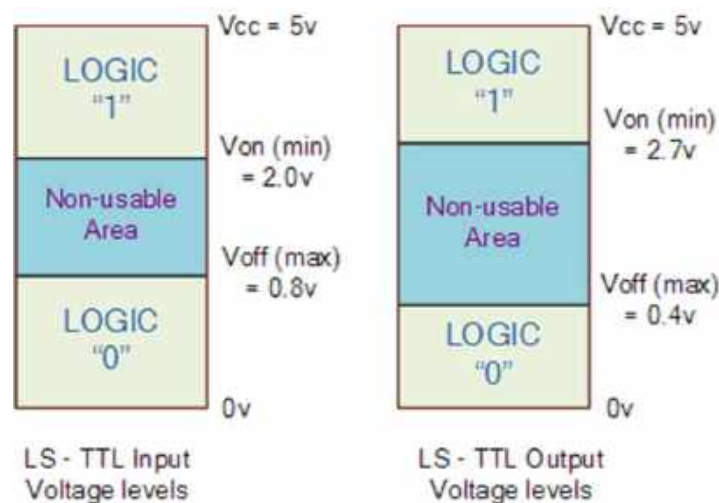
1. Prinsip kerja dari keluaran digital GPIO (*general purpose input output*).
2. Cara menggunakan fasilitas keluaran digital papan Arduino untuk berantarmuka dengan piranti digital eksternal.

INDIKATOR CAPAIAN PEMBELAJARAN

1. Mahasiswa mampu menjelaskan cara menggunakan fasilitas GPIO Arduino untuk menghasilkan keluaran digital dan mengerti prinsip kerjanya.
2. Mahasiswa mampu membuat aplikasi sederhana tentang penerapan GPIO untuk mengendalikan piranti keluaran secara digital.

TEORI SINGKAT

Keluaran digital merupakan fungsi dasar semua piranti digital seperti halnya masukan digital. GPIO Arduino memberikan fleksibilitas tinggi terhadap penggunaan pin apakah sebagai masukan atau sebagai keluaran dalam satu waktu. Pada dasarnya fungsi keluaran digital menghasilkan logika HIGH dan LOW yang mewakili tegangan 0 volts dan +5 volts. Semua pin GPIO pada papan Arduino UNO dapat dijadikan keluaran digital, termasuk pin masukan analog A0 – A5.



Gbr. 5.1 Karakteristik digital GPIO Arduino

Berdasarkan Gambar 5.1 yang menunjukkan karakteristik digital GPIO untuk masukan dan keluaran, dapat dilihat bahwa logika keluaran HIGH memiliki rentang tegangan antara 2.7 volts – 5 volts, sedangkan untuk logika keluaran LOW memiliki rentang tegangan antara 0 volts – 0.4 volts. Dan tegangan diantaranya adalah tidak dikenali.

ALAT DAN BAHAN PERCOBAAN

Pada sesi ini akan diperkenalkan tentang cara menghasilkan keluaran digital pada papan Arduino menggunakan fasilitas GPIO. Adapun alat dan bahan yang digunakan selama percobaan adalah:

1. Resistor 220 Ω

Bahan percobaan untuk mengatur tingkat kecerahan penampil LCD.

2. Multimeter Digital

Alat bantu pengukur tagangan yang akan digunakan untuk mengukur tegangan analog keluaran LPF.

3. LED (*light emitting diode*)

Bahan percobaan yang digunakan sebagai indikator logika keluaran digital.

4. LCD (*liquid crystal display*)

Bahan percobaan yang digunakan untuk menampilkan informasi visual dalam bentuk karakter huruf atau angka.

5. Papan Arduino UNO

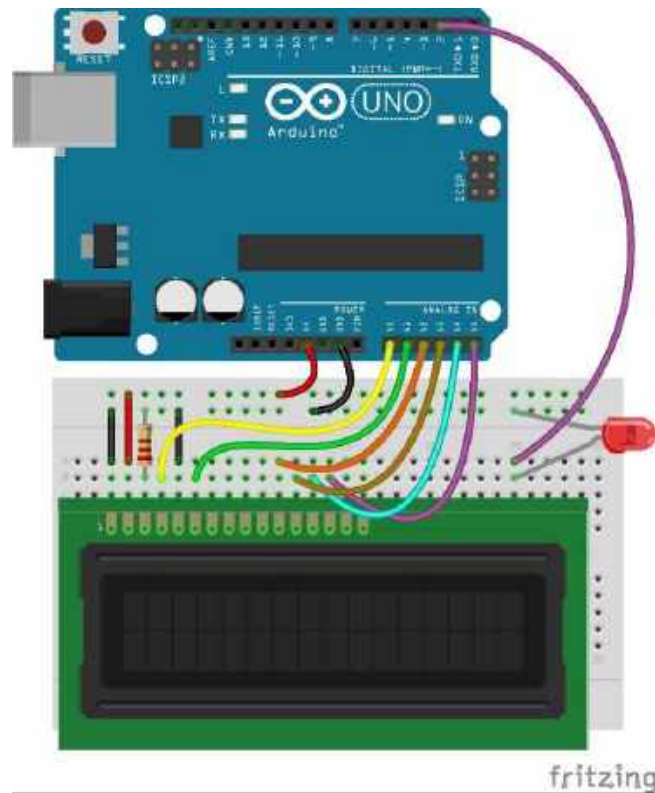
Bahan percobaan sebagai pengendali utama untuk menguji fungsi PWM dengan melibatkan aktivitas perancangan antarmuka perangkat keras dan pemrograman perangkat lunak.

LANGKAH PERCOBAAN

Percobaan fungsi keluaran analog dengan PWM oleh papan Arduino melibatkan dua aktifitas, yaitu perancangan antarmuka perangkat keras dan pemrograman perangkat lunak. Adapun langkah percobaannya adalah sebagai berikut:

1. Disain antarmuka perangkat keras

Percobaan ini melibatkan aktifitas menghubungkan pin keluaran digital pada papan Arduino UNO dengan piranti penampil secara benar. Pada percobaan ini akan memanfaatkan LED dan LCD sebagai indikator logika keluaran digital atau penampil informasi visual. Gambar 5.2 menunjukkan antarmuka perangkat keras sebagai bahan percobaan.



Gbr. 5.2 Antarmuka papan Arduino UNO dengan piranti penampil digital LCD.

2. Pemrograman perangkat lunak

Setelah perangkat keras selesai dirakit, selanjutnya membuat kode program untuk menggunakan akses keluaran digital penampil. Ketik contoh kode program berikut ini, kompilasi dan upload ke papan Arduino. Kemudian pengamatan dilakukan menggunakan piranti LCD.

```
#include <LiquidCrystal.h>           //library LCD
#define LED 2                        //definisi pin LED

LiquidCrystal lcd(A0, A1, A2, A3, A4, A5); //definisi fungsi pin output

void setup() {
  lcd.begin(16, 2);                 //inisialisasi LCD (16 kolom, 2 baris)
  pinMode(LED, OUTPUT);             //aktivasi fungsi pin sbg output
}

void loop() {
  lcd.setCursor(0, 0);              //kursor ke kolom 1 dari baris 1
  lcd.print("Praktikum TE-UTY");    //tampil karakter
  lcd.setCursor(0, 1);              //kursor ke kolom 1 dari baris 2
  lcd.print("Embedded System ");    //tampil karakter
  digitalWrite(LED, !digitalRead(LED)); //LED berkedip
  delay(1000);                      //jeda 1000 ms
}
```


3. Analisis Hasil Percobaan

Analisa hasil percobaan dilakukan berdasarkan pengukuran tegangan menggunakan multimeter pada pin keluaran digital LED untuk setiap logika keluaran digital. Hasil pengukuran disajikan dalam bentuk isian seperti yang ditunjukkan oleh Tabel 5.1.

Tabel 5.1 Hasil pengamatan level tegangan dari logika keluaran digital

Sample	Kondisi LED	$V_{\text{terukur pada LED}}$	$V_{\text{ideal Digital}}$	V_{selisih}
1	ON (HIGH)			
2	OFF (LOW)			

Selanjutnya berikan penjelasan mengapa terjadi selisih tegangan pada pin keluaran logika digital yang terhubung dengan LED.

TUGAS

Berikut ini deskripsi tugas yang harus anda lakukan terkait dengan tahap-tahap percobaan yang sudah dilakukan.

1. Membuat program COUNTER secara otomatis menggunakan fungsi delay (1 detik) dengan nilai cacahan antara 0 – 59, kemudian setiap nilai cacahan tampilkan pada penampil LCD.
2. Kompilasi dan upload kode program yang sudah anda buat, kemudian amati hasilnya.
3. Membuat laporan percobaan yang berisi foto antarmuka perangkat keras dan kode program yang berhasil dibuat dan berjalan dengan benar.

this page is intentionally left blank

VI. KOMUNIKASI DIGITAL UART

TUJUAN PEMBELAJARAN

Setelah melakukan praktik pada sesi ini, diharapkan mahasiswa mengetahui dan memahami beberapa hal berikut:

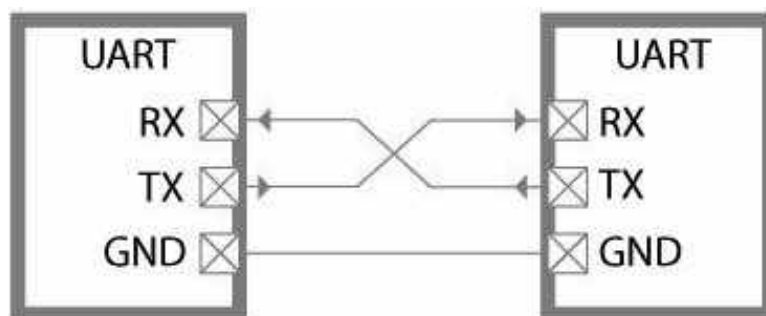
1. Prinsip kerja dari komunikasi UART (*Universal Asynchronous Receiver Transmitter*)
2. Cara menggunakan fasilitas UART pada papan Arduino untuk berkomunikasi dengan komputer atau piranti eksternal lain.

INDIKATOR CAPAIAN PEMBELAJARAN

1. Mahasiswa mampu menjelaskan cara menggunakan fasilitas UART Arduino untuk berkomunikasi data secara serial tak sinkron dan mengerti prinsip kerjanya.
2. Mahasiswa mampu membuat aplikasi sederhana tentang penerapan UART untuk berkomunikasi dengan piranti digital eksternal.

TEORI SINGKAT

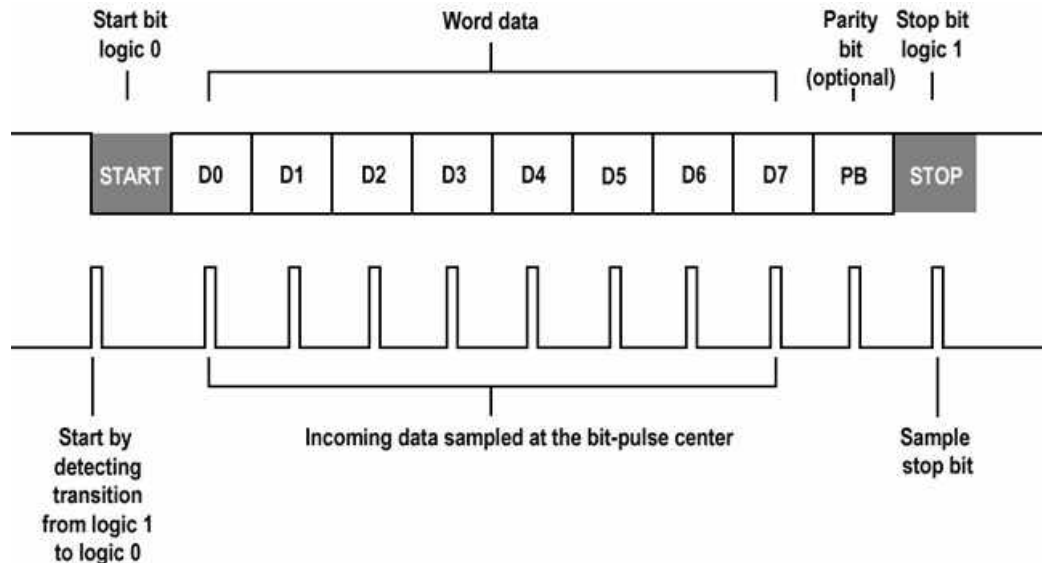
Komunikasi data serial sangat banyak diterapkan dalam teknologi digital modern karena faktor kecepatan dan kesederhanaan bus komunikasi datanya jika dibandingkan dengan komunikasi data digital secara parallel yang membutuhkan bus data besar dan tidak efektif untuk komunikasi jarak jauh. Komunikasi data digital secara serial terbagi menjadi dua jenis, komunikasi sinkron (*synchronous*) dan tak sinkron (*asynchronous*). Komunikasi data digital serial sinkron membutuhkan jalur khusus untuk sinkronisasi pewaktuan transmisi data serial sehingga pola komunikasi bersifat *half duplex*, yaitu komunikasi dilakukan secara bergantian. Sedangkan komunikasi data digital serial tak sinkron tidak memerlukan sinkronisasi pewaktuan transmisi data dengan asumsi bahwa dua bagian yang berkomunikasi memiliki kecepatan transmisi data (*baud rate*) yang sama, sehingga komunikasi dapat dilakukan secara bersamaan (*full duplex*).



Gbr. 6.1 Bus komunikasi digital serial tak sinkron UART

Gambar 6.1 di atas menunjukkan bus komunikasi serial tak sinkron antara dua piranti yang hanya memerlukan 3 jalur pengkawatn dalam bus komunikasi. Hal ini lah yang membuat komunikasi data serial tak sinkron banyak digunakan dari pada komunikasi data serial sinkron. Dengan karakteristik bus data komunikasi data tak sinkron seperti yang ditunjukkan oleh Gambar 6.1 memungkinkan untuk diterapkan pada komunikasi nir-kabel dalam jarak yang jauh.

Sistem pengendali mikro pada Arduino menerapkan komunikasi data serial jenis tak sinkron dengan pertimbangan fleksibilitas dan kesederhanaan pola komunikasinya. Komunikasi data serial tak sinkron memerlukan format data setiap paket transmisi agar dua piranti yang saling berkomunikasi bisa memahami paket data yang dikirim/diterima. Gambar 6.2 menunjukkan format paket transmisi setiap Byte data pada komunikasi serial tak sinkron.



Gbr. 6.2 Format paket transmisi data pada komunikasi serial tak sinkron.

Berdasarkan Gambar 6.2 dapat dipahami bahwa periode setiap bit data pada komunikasi tak sinkron harus disepakati oleh dua piranti yang berkomunikasi. Lebar bit data menentukan kecepatan transmisi serial yang dikenal dengan nama *baud rate*. Jika nilai baud rate tidak disepakati oleh dua piranti maka komunikasi data serial tidak akan berjalan dengan benar.

ALAT DAN BAHAN PERCOBAAN

Pada sesi ini akan diperkenalkan tentang cara melakukan komunikasi serial papan Arduino dengan PC/Laptop menggunakan fasilitas UART. Adapun alat dan bahan yang digunakan selama percobaan adalah:

1. Papan Arduino UNO

Bahan percobaan sebagai piranti digital utama untuk menguji komunikasi UART dengan PC/Laptop yang melibatkan aktivitas perancangan antarmuka perangkat keras dan pemrograman perangkat lunak.

LANGKAH PERCOBAAN

Percobaan komunikasi data serial UART oleh papan Arduino melibatkan dua aktifitas, yaitu perancangan antarmuka perangkat keras dan pemrograman perangkat lunak. Adapun langkah percobaannya adalah sebagai berikut:

1. Disain antarmuka perangkat keras

Antarmuka perangkat keras papan Arduino dengan PC/Laptop tidak ada yang harus dirakit, hal ini karena hanya menghubungkan papan Arduino dengan PC/Laptop melalui kabel komunikasi USB dan langkah ini memang pasti dilakukan untuk setiap sesi praktikum.

2. Pemrograman perangkat lunak

Percobaan yang akan dilakukan adalah melakukan komunikasi dua arah *full duplex*. Berikut ini contoh kode program anda harus buat untuk menguji komunikasi digital serial tak sinkron antara papan Arduino dan PC/Laptop.

```
String data_masuk = ""; //penampung teks
void setup(){
  Serial.begin(9600); //baudrate = 9600 bps
}

void loop(){
  if(Serial.available() > 0){ //ada data masuk?
    char ascii = Serial.read(); //ambil satu data
    if(ascii != '\r') //ada karakter "enter"?
      data_masuk = data_masuk + ascii; //jika tidak, kumpulkan
    else{
      if(data_masuk == "UTY") //keyword betul?
        Serial.println("Unggul"); //jika ya, jawab
      else
        Serial.println("???"); //jika tidak, jawab
      data_masuk = ""; //kosongkan penampung
    }
  }
}
```

Ketik contoh kode program di atas, kompilasi, dan upload ke papan Arduino. Kemudian pengamatan dilakukan menggunakan serial monitor pada Arduino IDE dan simpulkan.

TUGAS

Berikut ini deskripsi tugas yang harus anda lakukan terkait dengan tahap-tahap percobaan yang sudah dilakukan.

1. Membuat program yang merespon sebuah pertanyaan seperti yang sudah anda buat saat percobaan, minimal ada tiga pertanyaan.
2. Kompilasi dan upload kode program yang sudah anda buat, kemudian amati hasilnya.
3. Membuat laporan percobaan yang berisi foto antarmuka perangkat keras dan kode program yang berhasil dibuat dan berjalan dengan benar.

this page is intentionally left blank

VII. KOMUNIKASI SERIAL SINKRON - I²C

TUJUAN PEMBELAJARAN

Setelah melakukan praktik pada sesi ini, diharapkan mahasiswa mengetahui dan memahami beberapa hal berikut:

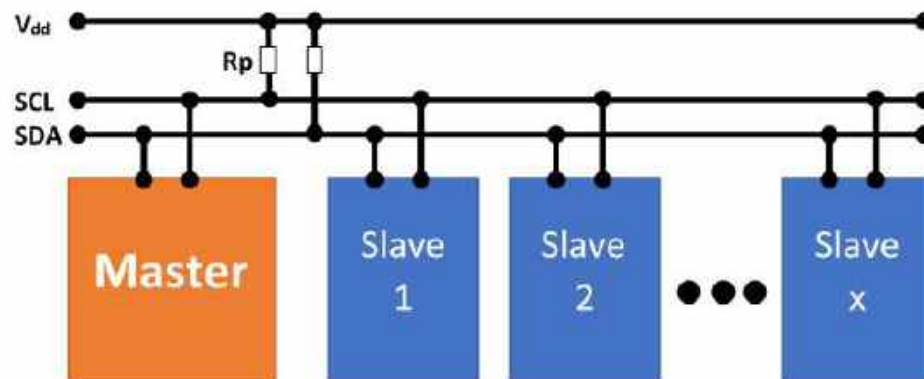
1. Prinsip kerja dari komunikasi serial sinkron I²C (*inter integrated circuit*).
2. Cara menggunakan fasilitas I²C pada papan Arduino untuk berkomunikasi dengan sensor atau piranti digital eksternal lain.

INDIKATOR CAPAIAN PEMBELAJARAN

1. Mahasiswa mampu menjelaskan cara menggunakan fasilitas I²C Arduino untuk berkomunikasi data secara serial sinkron dan mengerti prinsip kerjanya.
2. Mahasiswa mampu membuat aplikasi sederhana tentang penerapan I²C untuk berkomunikasi dengan sensor atau piranti digital eksternal.

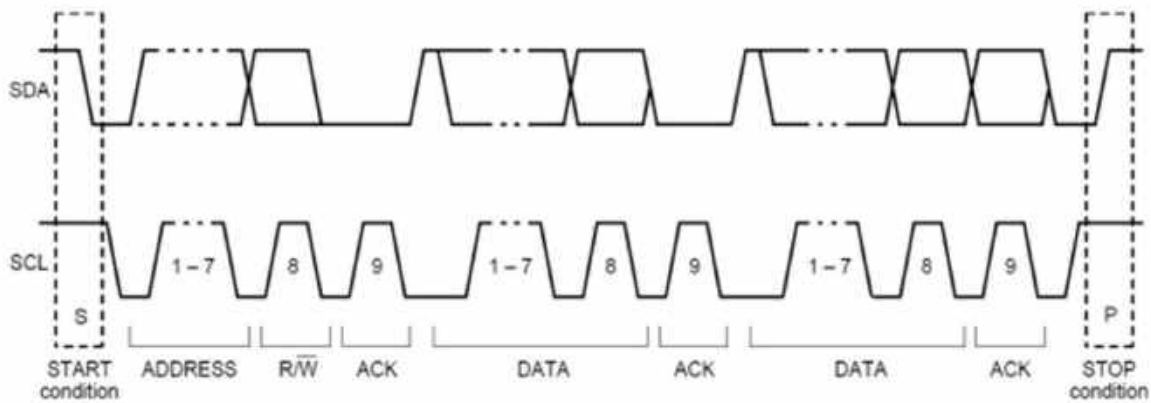
TEORI SINGKAT

Komunikasi data I²C merupakan komunikasi data digital serial secara sinkron, yaitu membutuhkan jalur sinkronisasi pewaktuan transmisi data serial. Karena bersifat komunikasi sinkron maka pola komunikasi antar piranti hanya bisa dilakukan secara *half duplex*. Kelebihan dari komunikasi serial sinkron I²C adalah memungkinkan untuk dilakukan komunikasi serial *multi-node*, yaitu komunikasi dengan banyak piranti dalam satu bus komunikasi yang sama. Hal ini berbeda dengan komunikasi tak sinkron UART yang bersifat *peer-to-peer* yaitu hanya bisa dilakukan oleh dua piranti saja. Selain itu, teknologi I²C tidak perlu pengaturan *baudrate* karena komunikasi dilakukan secara sinkron dengan kecepatan yang setiap saat bisa dirubah. Gambar 7.1 menunjukkan bus I²C untuk komunikasi beberapa piranti sekaligus.



Gbr. 7.1 Bus komunikasi I²C

Umumnya, antarmuka I²C digunakan Arduino untuk berkomunikasi dengan sensor-sensor. Dengan kelebihan bus I²C tersebut maka Arduino mampu membaca banyak sensor dalam satu bus saja, hal ini memberikan efektifitas perancangan sistem benam semakin tinggi. Dengan bus I²C maka akan dihemat pemakaian pin Arduino dan memudahkan perancangan sistem elektroniknya (misal: PCB). Seperti yang ditunjukkan oleh Gambar 7.1, pada bus komunikasi I²C terdapat 1-MASTER dan n-SLAVE. Bus komunikasi I²C terbagi menjadi dua: **SDA** (*serial data*) dan **SCL** (*synchronous clock*). SDA adalah bus transmisi data secara dua arah yang bersifat *half duplex*, sehingga hanya ada satu piranti saja yang boleh menggunakan dalam satu waktu. Sedangkan SCL adalah bus transmisi detak sinkronisasi yang bersifat satu arah, yaitu hanya piranti MASTER saja yang boleh menggunakan. Komunikasi hanya bisa dilakukan oleh MASTER – SLAVE saja dan komunikasi SLAVE – SLAVE tidak diperbolehkan.



Gbr. 7.2 Format paket transmisi data pada komunikasi serial sinkron I²C.

Format data komunikasi serial I²C berbeda dengan UART. Perbedaannya adalah pada keberadaan sinyal detak sinkronisasi seperti yang ditunjukkan oleh Gambar 7.2. Panjang data komunikasi I²C bisa bervariasi tergantung jenis pirantinya. Pada UART hanya dibatasi satu data saja untuk setiap paket pengiriman. Frekwensi detak bus I²C ada beberapa jenis, yaitu: *Fast Mode* (400Kbps), *Fast Mode+* (1Mbps), dan *High Speed Mode* (3.4Mbps). Dengan kecepatan setinggi itu maka bus komunikasi I²C umumnya sangat pendek, yang hanya menghubungkan beberapa piranti dalam satu papan rangkaian tercetak (*printed circuit board* – PCB).

Pada papan Arduino UNO, bus I²C terhubung dengan dua pin khusus yaitu: **A4** (SDA) dan **A5** (SCL). Dua pin tersebut secara umum adalah digunakan untuk masukan analog, namun memiliki fungsi lain yaitu sebagai bus komunikasi I²C.

ALAT DAN BAHAN PERCOBAAN

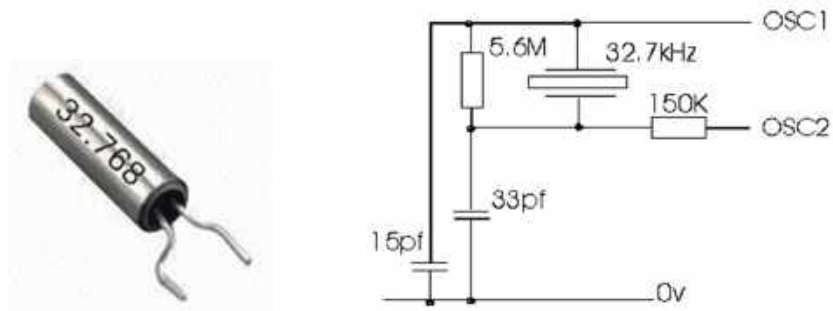
Pada sesi ini akan diperkenalkan tentang cara melakukan komunikasi serial papan Arduino dengan piranti RTC (*real time clock*) menggunakan fasilitas I²C. Adapun alat dan bahan yang digunakan selama percobaan adalah:

1. 2 Buah Resistor 4k7

Bahan percobaan yang digunakan sebagai resistor *pull up* bus komunikasi data serial I²C akar transmisi data bisa berjalan dengan baik.

2. Crystal 32.768KHz

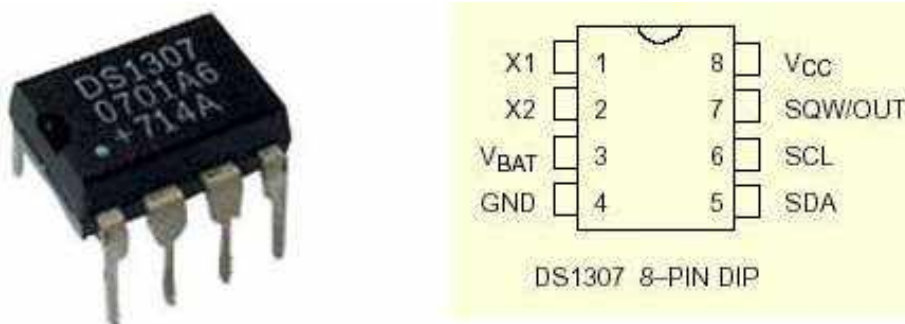
Bahan percobaan yang berfungsi sebagai komponen generator detak IC RTC DS1307. Gambar 7.3 menunjukkan bentuk fisik serta rangkaian operasi standar crystal pada RTC.



Gbr. 7.3 Bentuk fisik *crystal* RTC dan rangkaian operasionalnya.

3. IC RTC DS1307

Bahan percobaan yang berfungsi sebagai pewaktu digital dan menggunakan bus komunikasi data serial I²C. Data waktu yang diberikan oleh piranti ini meliputi: DD/MM/YY hh:mm:ss. Bentuk IC RTC DS1307 beserta tataletak kakinya ditunjukkan oleh Gambar 7.4 berikut.



Gbr. 7.4 IC Real Time Clock DS1307

4. Papan Arduino UNO

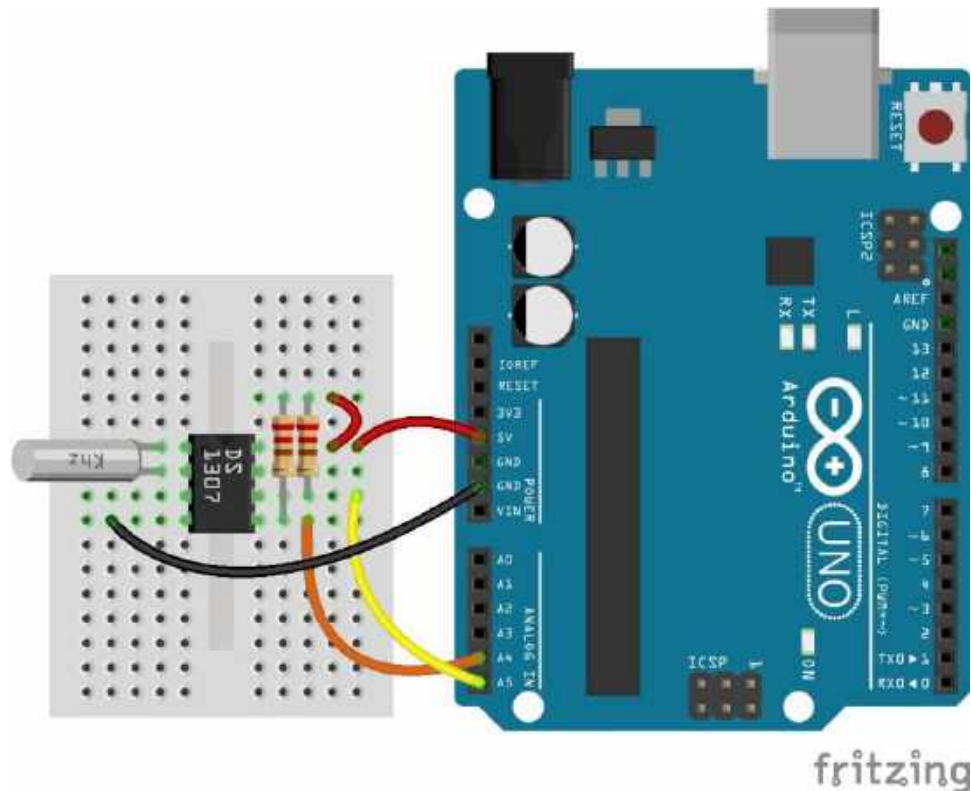
Bahan percobaan sebagai piranti digital utama untuk menguji komunikasi I²C dengan IC RTC DS1307 yang melibatkan aktivitas perancangan antarmuka perangkat keras dan pemrograman perangkat lunak.

LANGKAH PERCOBAAN

Percobaan komunikasi data serial I²C oleh papan Arduino melibatkan dua aktifitas, yaitu perancangan antarmuka perangkat keras dan pemrograman perangkat lunak. Adapun langkah percobaannya adalah sebagai berikut:

1. Disain antarmuka perangkat keras

Percobaan ini melibatkan aktifitas menghubungkan dua pin I²C pada papan Arduino UNO dengan piranti pewaktu DS1307. Gambar 7.5 menunjukkan antarmuka perangkat keras sebagai bahan percobaan.



Gbr. 7.5 Antarmuka papan Arduino UNO dengan IC RTC DS1307

2. Pemrograman perangkat lunak

Percobaan yang akan dilakukan adalah melakukan komunikasi I²C antara papan Arduino dengan IC RTC DS1307. Berikut ini contoh kode program anda harus buat untuk menguji komunikasi digital serial sinkron I²C dengan menggunakan Library: **Wire.h** dan **RTClib.h**

```

#include <Wire.h>           //definisi library I2C
#include "RTClib.h"        //definisi library RTC

RTC_DS1307 rtc;           //definisi object

void setup(){
  Serial.begin(9600);      //baudrate = 9600 bps
  if (!rtc.begin()){      //apakah RTC tidak terdeteksi?
    Serial.println("DS1307 tidak terdeteksi");
    while(1);
  }
  if (!rtc.isrunning()){ //apa RTC tidak aktif?
    Serial.println("RTC tidak aktif!");
    rtc.adjust(DateTime(2018, 1, 1, 0, 0, 0));
  }
}

void loop(){
  DateTime now = rtc.now(); //baca info waktu RTC
  Serial.print(now.day(), DEC); //kirim data tanggal
  Serial.print('/');
  Serial.print(now.month(), DEC); //kirim data bulan
  Serial.print('/');
  Serial.print(now.year(), DEC); //kirim data tahun
  Serial.print(" ");
  Serial.print(now.hour(), DEC); //kirim data jam
  Serial.print(':');
  Serial.print(now.minute(), DEC); //kirim data menit
  Serial.print(':');
  Serial.print(now.second(), DEC); //kirim data detik
  Serial.println();
  delay(1000);
}

```

Ketik contoh kode program di atas, kompilasi, dan upload ke papan Arduino. Kemudian pengamatan dilakukan menggunakan serial monitor pada Arduino IDE dan simpulkan.

TUGAS

Berikut ini deskripsi tugas yang harus anda lakukan terkait dengan tahap-tahap percobaan yang sudah dilakukan.

1. Membuat program **ALARM**, yaitu apabila waktu yang terbaca dari RTC sama dengan waktu yang diinginkan, maka kemudian kirimkan pemberitahuan melalui serial UART.
2. Kompilasi dan upload kode program yang sudah anda buat, kemudian amati hasilnya.
3. Membuat laporan percobaan yang berisi foto antarmuka perangkat keras dan kode program yang berhasil dibuat dan berjalan dengan benar.

this page is intentionally left blank

VIII. KOMUNIKASI SERIAL SINKRON - SPI

TUJUAN PEMBELAJARAN

Setelah melakukan praktik pada sesi ini, diharapkan mahasiswa mengetahui dan memahami beberapa hal berikut:

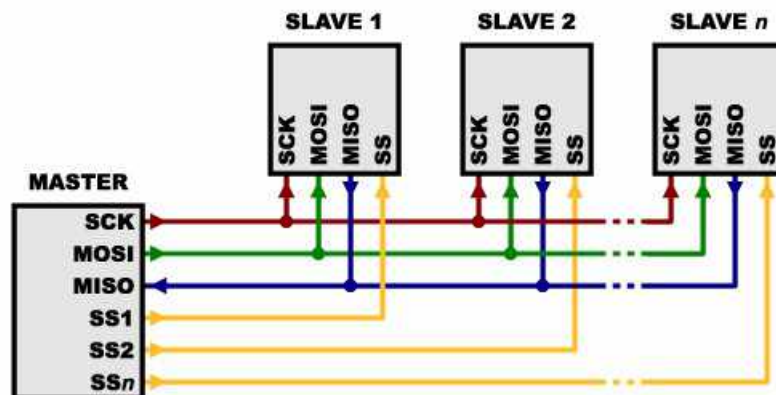
1. Prinsip kerja dari komunikasi serial sinkron SPI (*serial peripheral interface*).
2. Cara menggunakan fasilitas SPI pada papan Arduino untuk berkomunikasi dengan SD/MMC (*secure digital/multi media card*) atau piranti digital eksternal lain.

INDIKATOR CAPAIAN PEMBELAJARAN

1. Mahasiswa mampu menjelaskan cara menggunakan fasilitas SPI Arduino untuk berkomunikasi data secara serial sinkron dan mengerti prinsip kerjanya.
2. Mahasiswa mampu membuat aplikasi sederhana tentang penerapan SPI untuk menyimpan atau membaca informasi dari piranti memori SD/MMC.

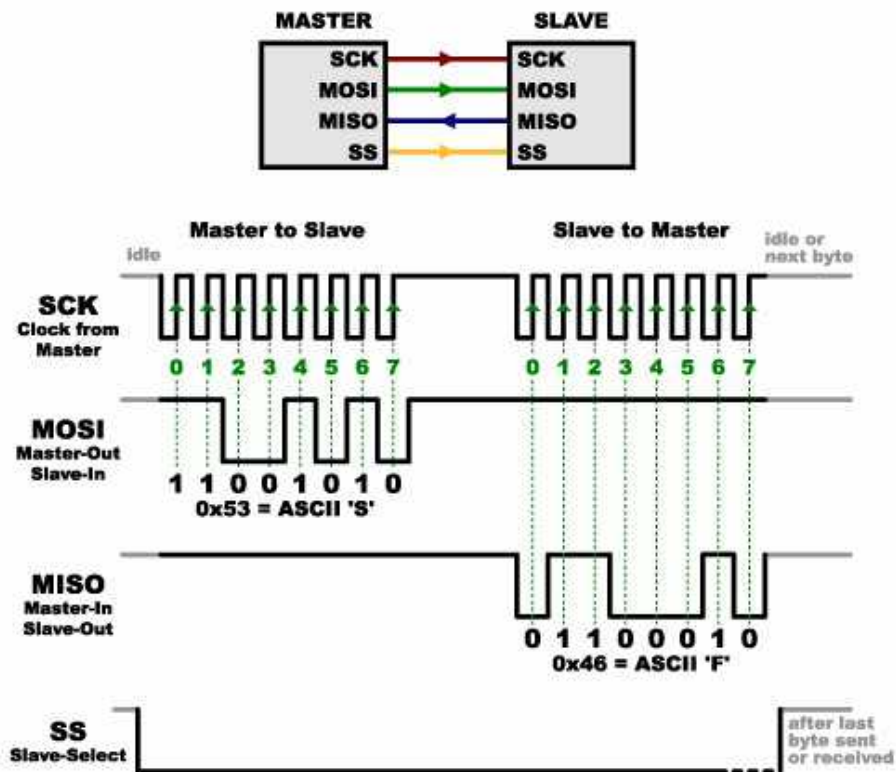
TEORI SINGKAT

Komunikasi data SPI merupakan komunikasi data digital serial secara sinkron sama seperti halnya I²C, yaitu membutuhkan jalur sinkronisasi pewaktuan transmisi data serial. Perbedaannya adalah pada jumlah bus data. Bus I²C hanya menggunakan satu jalur data (SDA) untuk berkomunikasi, sedangkan SPI menggunakan 2 jalur data (**MOSI** dan **MISO**) untuk berkomunikasi dua arah. Meskipun SPI menggunakan dua jalur komunikasi, komunikasi tetap dijalankan secara *half duplex* karena bersifat komunikasi sinkron. Kelebihan dari komunikasi serial sinkron SPI sama halnya dengan I²C yaitu memungkinkan untuk dilakukan komunikasi serial *multi-node*. Perbedaannya adalah SPI membutuhkan jalur tambahan untuk memilih *slave* (SS) sebanyak piranti yang terhubung pada bus komunikasi. Gambar 8.1 menunjukkan bus SPI untuk komunikasi beberapa piranti sekaligus.



Gbr. 8.1 Bus komunikasi SPI

Secara umum, antarmuka SPI juga digunakan Arduino untuk berkomunikasi dengan beberapa sensor sekaligus. Dengan kelebihan bus SPI tersebut maka Arduino mampu membaca banyak sensor dalam satu bus saja, hal ini memberikan efektifitas perancangan sistem benam semakin tinggi. Seperti yang ditunjukkan oleh Gambar 8.1, pada bus komunikasi SPI hanya terdapat 1-MASTER dan n-SLAVE. Berbeda dengan bus I²C yang memperbolehkan n-MASTER dan n-SLAVE dalam satu bus. Bus komunikasi SPI terbagi menjadi empat: **MOSI** (*master out – serial in*), **MISO** (*master in – serial out*), **SCK** (*serial clock*), dan **SS** (*slave select*). MOSI adalah bus transmisi data satu arah dari MASTER menuju SLAVE yang sudah dipilih berdasarkan pin SS, sehingga hanya ada satu SLAVE saja yang dapat berkomunikasi dalam satu waktu. MISO adalah bus transmisi data satu arah dari SLAVE yang terpilih berdasarkan pin SS menuju MASTER. Komunikasi data hanya bisa dilakukan oleh MASTER – SLAVE saja, sedangkan komunikasi SLAVE – SLAVE tidak diperbolehkan.



Gbr. 8.2 Format paket transmisi data pada komunikasi serial sinkron SPI.

Format data komunikasi serial SPI berbeda dengan I²C. Perbedaannya adalah pada paket data yang dikirimkan seperti yang ditunjukkan oleh Gambar 7.2. Paket data SPI tidak mengandung informasi alamat SLAVE, hal ini karena SLAVE sudah dipilih secara perangkat keras melalui jalur SS pada bus. Sehingga secara umum, format data SPI lebih sederhana dari pada I²C. Panjang data komunikasi SPI bisa bervariasi tergantung jenis pirantinya. Pada UART hanya dibatasi satu data saja untuk setiap paket pengiriman. Pada papan sistem Arduino, frekwensi detak bus SPI memiliki rentang yang lebar yaitu antara 125Kbps – 8Mbps.

Pada papan Arduino UNO, bus SPI terhubung dengan 4 pin khusus yaitu: **10 (SS)**, **11 (MOSI)**, **12 (MISO)**, dan **13 (SCK)**. Untuk memilih SLAVE tidak hanya terbatas pada pin 10 saja, namun bisa dialokasikan untuk pin digital yang lain sebanyak yang diperlukan.

ALAT DAN BAHAN PERCOBAAN

Pada sesi ini akan diperkenalkan tentang cara melakukan komunikasi serial papan Arduino dengan piranti SD/MMC menggunakan fasilitas SPI. Adapun alat dan bahan yang digunakan selama percobaan adalah:

1. SparkFun microSD

Bahan percobaan yang berfungsi sebagai penyimpan data digital berkapasitas besar.

2. Papan Arduino UNO

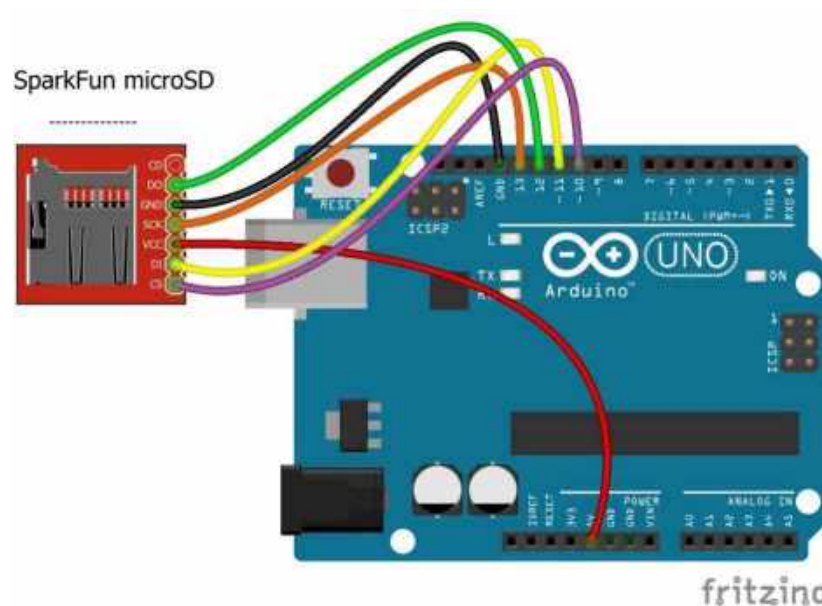
Bahan percobaan sebagai piranti digital utama untuk menguji komunikasi SPI dengan piranti memory microSD.

LANGKAH PERCOBAAN

Percobaan komunikasi data serial SPI oleh papan Arduino melibatkan dua aktifitas, yaitu perancangan antarmuka perangkat keras dan pemrograman perangkat lunak. Adapun langkah percobaannya adalah sebagai berikut:

1. Disain antarmuka perangkat keras

Percobaan menghubungkan papan Arduino UNO dengan piranti mikroSD. Gambar 8.3 menunjukkan antarmuka perangkat keras sebagai percobaan.



Gbr. 8.3 Antarmuka papan Arduino UNO dengan memori microSD

2. Pemrograman perangkat lunak

Percobaan yang akan dilakukan adalah melakukan komunikasi SPI antara papan Arduino dengan microSD. Berikut ini contoh kode program anda harus buat untuk menguji komunikasi digital serial sinkron SPI dengan menggunakan Library: **SPI.h** dan **SD.h**

```
#include <SPI.h>
#include <SD.h>

#define SS 10

sd2Card  card;
sdVolume volume;
sdFile   root;

void setup() {
  Serial.begin(9600);           //baudrate = 9600bps
  Serial.println("Inisialisasi microSD");
  if (!card.init(SPI_HALF_SPEED,SS)) {           //apa kartu terdeteksi?
    Serial.println("inisialisasi gagal");       //jika tidak
    while (1);
  }
  else{
    Serial.println("inisialisasi berhasil");     //jika ya
    Serial.print("Card type: ");               //tampilkan informasi
    switch (card.type()) {
      case SD_CARD_TYPE_SD1:
        Serial.println("SD1");
        break;
      case SD_CARD_TYPE_SD2:
        Serial.println("SD2");
        break;
      case SD_CARD_TYPE_SDHC:
        Serial.println("SDHC");
        break;
      default:
        Serial.println("Unknown");
    }
  }
}

void loop() {} //tidak melakukan apa-apa
```

Ketik contoh kode program di atas, kompilasi, dan upload ke papan Arduino. Kemudian pengamatan dilakukan menggunakan serial monitor pada Arduino IDE dan simpulkan.

TUGAS

Berikut ini deskripsi tugas yang harus anda lakukan terkait dengan tahap-tahap percobaan yang sudah dilakukan.

1. Membuat file dengan nama *nama_file.txt* menggunakan notepad pada PC/Laptop, kemudian simpan file tersebut pada microSD. Perlu diperhatikan bahwa *nama_file* tidak boleh lebih dari 8 karakter, contoh = "log.txt" seperti pada contoh kode program.
2. Melakukan proses menyimpan data ke dalam file yang sudah dibuat pada microSD menggunakan papan Arduino. Ketik contoh kode program berikut, kompilasi, dan upload ke papan Arduino.

```
#include <SPI.h>
#include <SD.h>
#define SS 10

File myFile;
void setup() {
  Serial.begin(9600); //baudrate = 9600bps
  Serial.println("Inisialisasi microSD...");
  if (!SD.begin(SS)) { //apa SD terdeteksi?
    Serial.println("inisialisasi gagal"); //jika tidak
    while (1);
  }
  else{
    Serial.println("inisialisasi sukses"); //jika ya
    Serial.println("menyimpan file...");
    myFile = SD.open("log.txt", FILE_WRITE); //mode tulis
    if (myFile) { //apa file tersedia?
      myFile.println("Nama = Joko Widodo"); //jika ya
      myFile.println("Alamat = Jakarta");
      myFile.println("Pekerjaan = Presiden");
      myFile.close();
      Serial.println("penyimpanan selesai");
    }
    else{
      Serial.println("gagal membuka file"); //jika tidak
    }
  }
}
void loop() {} //tidak melakukan apa-apa
```

3. Melakukan proses pembacaan data dari file yang tersimpan dan terisi pada microSD menggunakan papan Arduino. Ketik contoh kode program berikut, kompilasi, dan upload ke papan Arduino.

```
#include <SPI.h>
#include <SD.h>
#define SS 10

File myFile;
void setup() {
  Serial.begin(9600); //baudrate = 9600bps
  Serial.println("Inisialisasi microSD...");
  if (!SD.begin(SS)) { //apa SD terdeteksi?
    Serial.println("inisialisasi gagal"); //jika tidak
    while (1);
  }
  else {
    Serial.println("inisialisasi sukses"); //jika ya
    Serial.println("membaca isi file...");
    myFile = SD.open("log.txt"); //mode baca
    if (myFile) { //apa file tersedia?
      Serial.println("isi file:"); //jika ya, tampilkan
      while (myFile.available()) {
        Serial.write(myFile.read());
      }
      myFile.close();
    }
    else {
      Serial.println("gagal membuka file"); //jika tidak
    }
  }
}
void loop() {} //tidak melakukan apa-apa
```

4. Ulangi percobaan dengan mengisikan informasi yang berbeda pada file di dalam microSD.
5. Membuat laporan percobaan yang berisi foto antarmuka perangkat keras dan kode program yang berhasil dibuat dan berjalan dengan benar.

this page is intentionally left blank

IX. REAL TIME OPERATING SYSTEM (RTOS)

TUJUAN PEMBELAJARAN

Setelah melakukan praktik pada sesi ini, diharapkan mahasiswa mengetahui dan memahami beberapa hal berikut:

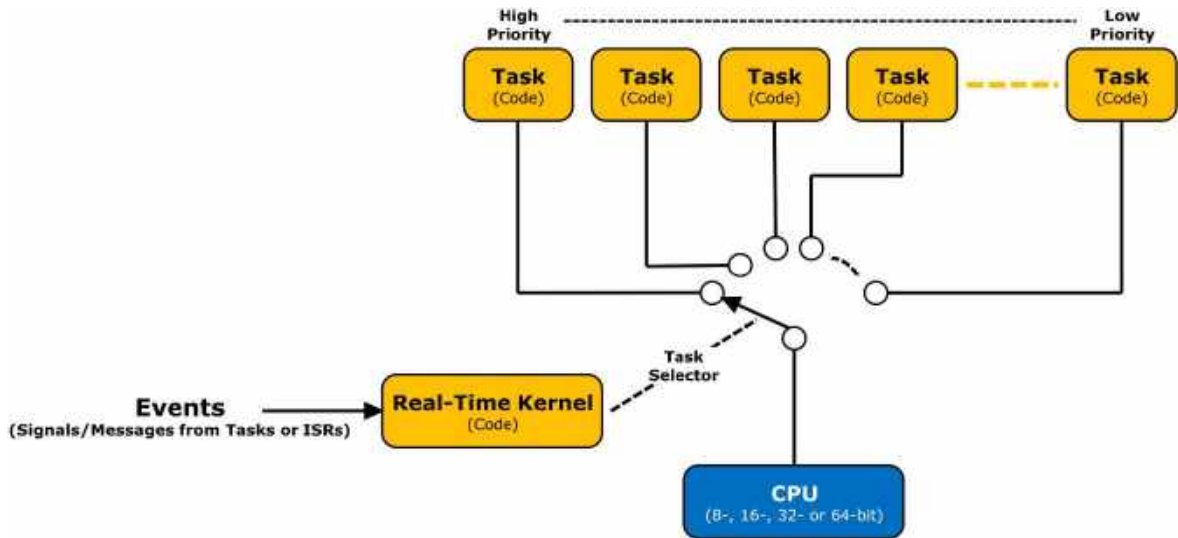
1. Pengertian dari RTOS pada sistem benam.
2. Cara menggunakan RTOS pada papan Arduino untuk menyelesaikan beberapa tugas dalam waktu yang hampir bersamaan (*multi tasking*).

INDIKATOR CAPAIAN PEMBELAJARAN

1. Mahasiswa mampu menjelaskan cara menggunakan menggunakan RTOS pada sistem benam berbasis Arduino.
2. Mahasiswa mampu membuat aplikasi sederhana tentang penerapan RTOS untuk mengerjakan beberapa fungsi sekaligus.

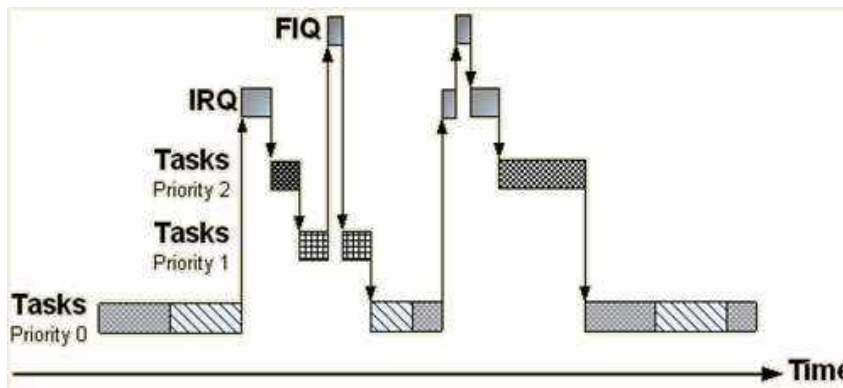
TEORI SINGKAT

RTOS atau sistem operasi waktu nyata adalah sebuah sistem operasi yang dirancang untuk melayani beberapa tugas pemrosesan data masuk/keluar secara real-time tanpa penundaan, dikenal dengan istilah *multi-tasking*. Umumnya sistem operasi ini dijalankan pada platform sistem benam (*embedded system*). Pada dasarnya, sebuah *microprocessor core* tidak mampu mengerjakan beberapa tugas secara bersamaan (*concurrent*) karena memang sifat eksekusi perintahnya adalah berurutan (*sequence*). Kelemahan dari metode sekuensial tradisional adalah adanya waktu tunda eksekusi suatu program karena menunggu eksekusi program lain yang belum selesai. Jika terdapat banyak program yang harus dilakukan oleh *microprocessor* maka konsep sekuensial model ini sudah tidak bisa andalkan. RTOS hadir dengan kernel *real-time scheduler* yang akan menjadwalkan eksekusi beberapa pekerjaan secara bergiliran dengan kecepatan tinggi, sehingga terkesan adanya eksekusi suatu program secara simultan meskipun kenyataannya program tersebut mengalami jeda eksekusi beberapa kali karena *microprocessor* mengeksekusi program lain. RTOS pada sebuah sistem benam tidak menjalankan pemrosesan secara parallel karena memang arsitektur sistem benam tidak mendukungnya, namun dengan melakukan penjadwalan eksekusi dari beberapa pekerjaan maka pekerjaan-pekerjaan tersebut dapat diselesaikan seolah-olah tanpa jeda. Prinsip kerja dari *scheduler* RTOS adalah membagi slot waktu eksekusi (*tick time*) untuk tiap-tiap program dari beberapa program yang dijalankan. Gambar 9.1 mendeskripsikan prinsip kerja sebuah *scheduler* RTOS yang menjadwalkan eksekusi setiap program oleh CPU bergiliran secara cepat.



Gbr. 9.1 Prinsip kerja scheduler RTOS

Setiap program yang dijadwalkan oleh scheduler untuk dieksekusi diberikan prioritas dari HIGH ke LOW, hal tersebut untuk menentukan pola penggiliran sesuai dengan prioritasnya. Disamping menjalankan penjadwalan, RTOS juga menerima permintaan interupsi baik dari perangkat keras maupun dari perangkat lunak yang meminta layanan. Ilustrasi penggiliran RTOS dan pelayanan interupsi dapat dilihat pada Gambar 9.2 di bawah ini.



Gbr. 9.2 Pola penjadwalan RTOS terhadap Task dan Interrupt.

Terdapat banyak RTOS yang sudah dirancang untuk diterapkan pada sistem benam, seperti **FreeRTOS**. Beberapa orang telah melakukan porting freeRTOS agar dapat diterapkan pada sistem benam berbasis Arduino, yaitu menyesuaikan resource MCU (microcontroller unit) yang digunakan oleh Arduino agar mampu menjalankan freeRTOS dengan baik dan efisien ditinjau dari penggunaan ruang memory ROM dan RAM (*footprint*) oleh kernel freeRTOS.

ALAT DAN BAHAN PERCOBAAN

Pada sesi ini akan diperkenalkan tentang cara menggunakan RTOS untuk menjalankan beberapa fungsi secara bersamaan. Adapun alat dan bahan yang digunakan selama percobaan adalah:

1. 3 buah LED (*light emitting diode*)

Bahan percobaan yang berfungsi sebagai indikator proses secara visual.

2. Papan Arduino UNO

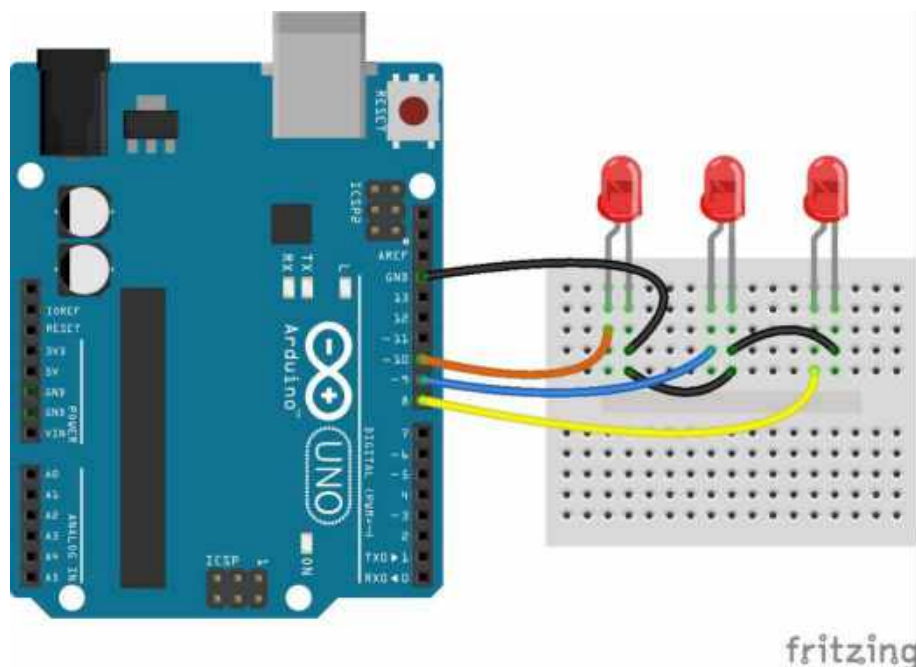
Bahan percobaan sebagai piranti digital utama untuk menguji kinerja RTOS.

LANGKAH PERCOBAAN

Percobaan penerapan RTOS pada papan Arduino melibatkan dua aktifitas, yaitu perancangan antarmuka perangkat keras dan pemrograman perangkat lunak. Adapun langkah percobaannya adalah sebagai berikut:

1. Disain antarmuka perangkat keras

Percobaan menghubungkan papan Arduino UNO dengan indikator LED. Gambar 9.3 menunjukkan antarmuka perangkat keras sebagai percobaan.



Gbr. 9.3 Antarmuka papan Arduino UNO dengan 3 buah LED

2. Pemrograman perangkat lunak

Percobaan yang akan dilakukan adalah menerapkan RTOS pada papan Arduino untuk menghidup-matikan tiga LED dengan frekwensi yang berbeda. Berikut ini contoh kode program harus anda buat untuk menguji performa RTOS dengan menggunakan Library: **Arduino_FreeRTOS.h**

```
#include <Arduino_FreeRTOS.h>
#include <semphr.h>

#define LED1 8
#define LED2 9
#define LED3 10

SemaphoreHandle_t xSerialSemaphore;

void Task_LED1(void *pvParameters);
void Task_LED2(void *pvParameters);
void Task_LED3(void *pvParameters);

void setup() {
    Serial.begin(9600);
    if (xSerialSemaphore == NULL){
        xSerialSemaphore = xSemaphoreCreateMutex();
        if ((xSerialSemaphore) != NULL)
            xSemaphoreGive((xSerialSemaphore));
    }
    xTaskCreate(Task_LED1, (const portCHAR*) "T1", 128, NULL, 1, NULL);
    xTaskCreate(Task_LED2, (const portCHAR*) "T2", 128, NULL, 1, NULL);
    xTaskCreate(Task_LED3, (const portCHAR*) "T3", 128, NULL, 1, NULL);
}

void loop(){}

void Task_LED1(void *pvParameters __attribute__((unused))){
    pinMode(LED1, OUTPUT);
    for (;;) {
        digitalWrite(LED1, !digitalRead(LED1));
        vTaskDelay(30);
    }
}

void Task_LED2(void *pvParameters __attribute__((unused))){
    pinMode(LED2, OUTPUT);
    for (;;) {
        digitalWrite(LED2, !digitalRead(LED2));
        vTaskDelay(60);
    }
}

void Task_LED3(void *pvParameters __attribute__((unused))){
    pinMode(LED3, OUTPUT);
    for (;;) {
        digitalWrite(LED3, !digitalRead(LED3));
        vTaskDelay(90);
    }
}
```


Ketik contoh kode program di atas, kompilasi, dan upload ke papan Arduino. Kemudian lakukan pengamatan secara visual dan simpulkan.

TUGAS

Berikut ini deskripsi tugas yang harus anda lakukan terkait dengan tahap-tahap percobaan yang sudah dilakukan.

1. Membuat aplikasi multi-tasking yang berisi program menyalakan 3 buah LED dengan pola nyala yang berbeda-beda, dan program pengiriman data serial secara periodik. Sehingga terdapat 4 tugas yang harus dijalankan oleh Arduino secara Bersama-sama.
2. Kompilasi dan upload kode program yang sudah anda buat, kemudian amati hasilnya.
3. Membuat laporan percobaan yang berisi foto antarmuka perangkat keras dan kode program yang berhasil dibuat dan berjalan dengan benar.

this page is intentionally left blank

X. PROYEK-1: MENGUKUR DAN MENAMPILKAN DATA SUHU

TUJUAN PEMBELAJARAN

Sesi ini akan dilakukan perancangan proyek instrumentasi yaitu pengukuran suhu dan menampilkan data hasil pengukuran melalui piranti penampil.

INDIKATOR CAPAIAN PEMBELAJARAN

Mahasiswa mampu merancang proyek instrumentasi sederhana untuk pengukuran suhu ruang menggunakan sensor analog LM35 dan menampilkan hasilnya pada LCD serta memantaunya melalui terminal serial.

TEORI SINGKAT

Proyek ini merupakan gabungan hasil percobaan yang telah anda lakukan sebelumnya, yaitu bagian antarmuka masukan analog dan antarmuka keluaran digital. Percobaan yang akan anda lakukan adalah merancang dan membuat sistem instrumentasi ukur untuk memantau besaran alamiah berupa suhu dan menampilkan informasi hasil pengukuran yang akurat pada penampil digital.

ALAT DAN BAHAN PERCOBAAN

Pada sesi ini akan dibuat sistem instrumentasi ukur suhu secara digital. Adapun alat dan bahan yang digunakan selama percobaan adalah:

1. Resistor 220Ω

Bahan percobaan yang digunakan untuk mengatur tingkat kecerahan penampil LCD.

2. LED

Bahan percobaan yang digunakan sebagai indikator visual suatu proses.

3. IC LM35

Bahan percobaan yang berfungsi sebagai sensor pengukur suhu dengan keluaran tegangan analog.

4. LCD 16x2

Bahan percobaan yang berfungsi sebagai penampil informasi secara digital yang mampu menampilkan 32 karakter ASCII dalam dua baris.

5. Papan Arduino UNO

Bahan percobaan yang digunakan sebagai piranti digital utama untuk menjalankan fungsi pembacaan masukan analog dari LM35 secara akurat, dan menampilkan informasi hasil ukur ke penampil LCD.

LANGKAH PERCOBAAN

Percobaan kali ini merupakan gabungan percobaan-percobaan yang telah anda lakukan sebelumnya. Adapun langkah percobaan yang harus anda lakukan adalah:

1. Melakukan antarmuka papan Arduino UNO dengan piranti sensor suhu LM35.
2. Melakukan antarmuka papan Arduino UNO dengan piranti LCD 16x2
3. Menerapkan RTOS dalam penyusunan kode program.
4. Membuat kode program yang terdiri dari dua tugas (*task*):
 - a. Menyalakan indikator berkedip secara berkala dengan frekwensi 2Hz.
 - b. Membaca variabel suhu kemudian menampilkannya pada LCD dan juga mengirimkannya melalui terminal serial UART.
5. Kompilasi kode program hingga tidak ada kesalahan sintaksis, kemudian upload ke papan Arduino UNO.
6. Amati hasil yang ditampilkan, dan buat kesimpulan dari percobaan yang telah dilakukan.

TUGAS

Berikut ini deskripsi tugas yang harus anda lakukan terkait dengan tahap-tahap percobaan yang sudah dilakukan.

1. Membuat materi presentasi terkait rancangan proyek yang sudah berhasil dibuat.
2. Presentasikan di depan kelas.
3. Membuat laporan percobaan yang berisi rancangan antarmuka perangkat keras dan kode program yang berhasil dibuat dan berjalan dengan benar.

this page is intentionally left blank

XI. PROYEK-2: DATA LOGGER SUHU

TUJUAN PEMBELAJARAN

Sesi ini akan dilakukan perancangan proyek *data logger*, yaitu menyimpan hasil pengukuran suhu ke dalam memori permanen berdasarkan waktu (*time stamp*).

INDIKATOR CAPAIAN PEMBELAJARAN

Mahasiswa mampu merancang proyek *data logger* untuk menyimpan hasil pengukuran suhu ruang menggunakan sensor analog LM35 dan menyimpannya pada piranti memori microSD berdasarkan waktu yang diperoleh dari piranti RTC DS1307.

TEORI SINGKAT

Proyek ini merupakan gabungan hasil percobaan yang telah anda lakukan sebelumnya, yaitu percobaan antarmuka masukan analog, komunikasi I²C, dan komunikasi SPI. Percobaan yang akan anda lakukan adalah merancang dan membuat sistem *data logger* nilai suhu. Definisi data logger adalah sistem perekam informasi digital secara permanen yang melibatkan sensor – pewaktu digital – sistem benam – memori permanen berkapasitas besar. Setiap data hasil pengukuran akan dicatat ke dalam memori Bersama-sama dengan data waktu (DD/MM/YYYY hh:mm:ss). Isu utama pada data logger ada dua hal: masa hidup dan kapasitas memori. Masa hidup logger terkait dengan penggunaan baterai yang harus mampu bertahan lama agar logger dapat merekam data sebanyak mungkin, sehingga harus disediakan kapasitas memori yang juga mendukung untuk pemakaian dalam waktu yang lama. Dengan demikian agar energi listrik logger mampu bertahan lama maka sistem logger baik perangkat keras maupun perangkat lunak harus dirancang untuk mampu bekerja pada mode rendah konsumsi daya (*low power*).

ALAT DAN BAHAN PERCOBAAN

Pada sesi ini akan dibuat sistem data logger suhu. Adapun alat dan bahan yang digunakan selama percobaan adalah:

1. 2 buah Resistor 4.7k Ω

Bahan percobaan yang digunakan sebagai resistor *pull up* bus komunikasi I²C.

2. IC LM35

Bahan percobaan yang berfungsi sebagai sensor pengukur suhu dengan keluaran tegangan analog.

3. IC RTC DS1307

Bahan percobaan yang berfungsi sebagai penyimpan informasi waktu secara digital.

4. SparkFun microSD

Bahan percobaan yang digunakan untuk menyimpan informasi pengukuran dengan kapasitas besar.

5. Papan Arduino UNO

Bahan percobaan sebagai piranti digital utama untuk menjalankan fungsi pembacaan masukan analog dari sensor suhu LM35, pembacaan waktu dengan RTC DS1307, dan penyimpanan data pada memori microSD.

LANGKAH PERCOBAAN

Percobaan kali ini merupakan gabungan percobaan-percobaan yang telah anda lakukan sebelumnya. Adapun langkah percobaan yang harus anda lakukan adalah:

1. Melakukan antarmuka papan Arduino UNO dengan piranti sensor suhu LM35.
2. Melakukan antarmuka papan Arduino UNO dengan piranti RTC DS1307.
3. Melakukan antarmuka papan Arduino UNO dengan piranti memori microSD.
4. Membuat kode program untuk membaca variabel suhu, membaca waktu (DD/MM/YYYY hh:mm:ss), dan menyimpan informasi kedalam pengukuran kedalam memori microSD. Adapun kriteria penyimpanan adalah:
 - a. Format informasi *data logger*: [DD/MM/YYYY hh:mm:ss suhu]
 - b. Format file tempat menampung informasi data logger: [*nama.txt*] dengan maksimal karakter nama adalah 8.
5. Kompilasi kode program hingga tidak ada kesalahan sintaksis, kemudian upload ke papan Arduino UNO.
6. Amati hasil yang ditampilkan, dan buat kesimpulan dari percobaan yang telah dilakukan.

TUGAS

Berikut ini deskripsi tugas yang harus anda lakukan terkait dengan tahap-tahap percobaan yang sudah dilakukan.

1. Membuat materi presentasi terkait rancangan proyek yang sudah berhasil dibuat.
2. Presentasikan di depan kelas.
3. Membuat laporan percobaan yang berisi rancangan antarmuka perangkat keras dan kode program yang berhasil dibuat dan berjalan dengan benar.

this page is intentionally left blank

XII. PROYEK-3: PENGENDALI SUHU OTOMATIS

TUJUAN PEMBELAJARAN

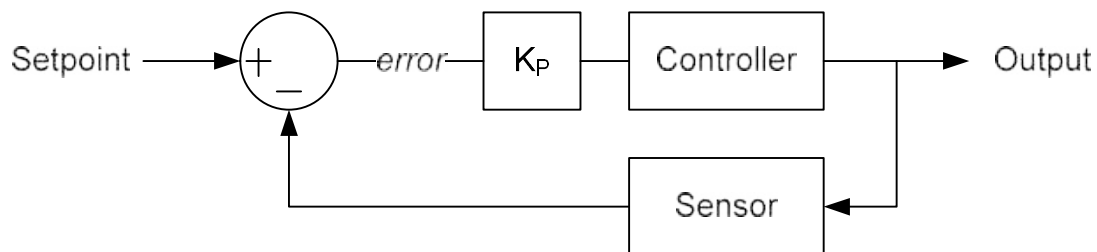
Sesi ini akan dilakukan perancangan proyek pengendali suhu otomatis dengan metode proporsional, yaitu bagaimana mengendalikan suhu ruang secara optimal dengan memanfaatkan sensor suhu dan aktuator pendinginan.

INDIKATOR CAPAIAN PEMBELAJARAN

Mahasiswa mampu merancang sistem pengendali suhu secara otomatis dengan mengukur suhu ruangan menggunakan sensor analog LM35 dan berusaha menjaganya agar tetap stabil pada nilai yang dikehendaki menggunakan aktuator pendinginan berupa kipas.

TEORI SINGKAT

Proyek ini merupakan gabungan hasil percobaan yang telah anda lakukan sebelumnya, yaitu percobaan antarmuka masukan analog, dan antarmuka keluaran analog PWM. Percobaan yang akan anda lakukan adalah merancang dan membuat sistem kendali suhu otomatis dengan metode proporsional. Model kendali proporsional berbeda dengan kendali on-off yang menggunakan satu batas nilai (*threshold*) untuk menentukan apakah aktuator ON atau OFF. Model kendali proporsional adalah model kendali umpan tertutup (*close-loop control*) yang menggunakan umpan balik sensor suhu untuk mendapatkan nilai error. Nilai error yang diterima secara proporsional akan menentukan penguatan kendali aktuator. Semakin besar nilai error maka semakin besar penguatan kendalinya. Gambar 12.1 menunjukkan model kendali proporsional.



Gbr. 12.1 Model kendali proporsional umpan tertutup

Model kendali proporsional akan menjaga nilai keluaran (*output*) agar sama seperti nilai yang diinginkan (*setpoint*). Semakin besar nilai K_P maka karakteristik pengendaliannya semakin mudah untuk stabil, namun akan memperbesar kesalahan waku tunaknya (*steady state error*). Sebaliknya, jika K_P semakin kecil maka transient response nya semakin lama atau lambat mencapai kestabilan. Sehingga output kendali model proporsional adalah perkalian antara nilai *error* dengan K_P , $P = K_P * ERROR$.

Implementasi model kendali proporsional pada Arduino melibatkan penggunaan keluaran analog PWM yang juga mampu memberikan variasi nilai secara proporsional pula. Nilai keluaran P tidak ada batasnya, sedangkan nilai keluaran PWM Arduino memiliki batas maksimal (=255). Sehingga diperlukan *scaling* atau *thresholding* terhadap keluaran kendali P disesuaikan dengan nilai keluaran analog PWM Arduino.

ALAT DAN BAHAN PERCOBAAN

Pada sesi ini akan dibuat sistem kendali suhu otomatis. Adapun alat dan bahan yang digunakan selama percobaan adalah:

1. DC Motor (*Air Blower*)

Bahan percobaan yang digunakan sebagai aktuator pendinginan suhu. Piranti ini bekerja pada arus DC dengan tegangan operasi +12 volts. Gambar 12.2 menunjukkan bentuk fisik dari DC *air blower*.



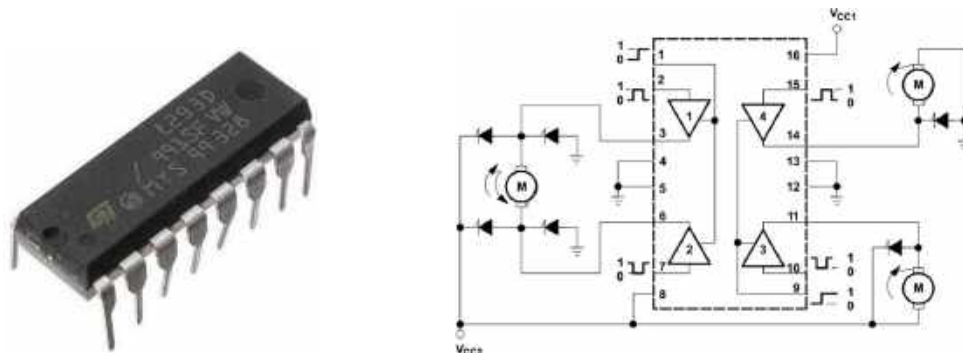
Gbr. 12.2 Bentuk fisik dari DC motor untuk *air blower*.

2. IC LM35

Bahan percobaan yang berfungsi sebagai sensor pengukur suhu dengan keluaran tegangan analog.

3. IC L293D

Bahan percobaan yang berfungsi sebagai penggerak motor DC. Piranti ini mampu menggerakkan DC motor arus rendah (maksimal = 2A). Gambar 12.3 menunjukkan bentuk fisik dan konfigurasi mode kerja L293D dalam menggerakkan DC motor.



Gbr. 12.3 Bentuk IC L293D dan konfigurasi beberapa mode kerjanya.

TUGAS

Berikut ini deskripsi tugas yang harus anda lakukan terkait dengan tahap-tahap percobaan yang sudah dilakukan.

1. Membuat materi presentasi terkait rancangan proyek yang sudah berhasil dibuat.
2. Presentasikan di depan kelas.
3. Membuat laporan percobaan yang berisi rancangan antarmuka perangkat keras dan kode program yang berhasil dibuat dan berjalan dengan benar.

this page is intentionally left blank

@2019

Diterbitkan oleh:

Universitas Teknologi Yogyakarta

Jl. Siliwangi, Jombor, Sleman, Yogyakarta

Email : publikasi@uty.ac.id

Website : uty.ac.id