

MODUL 4

Komunikasi kabel menggunakan UART, I2C dan SPI dengan Arduino UNO

1. TUJUAN

- 1.1 Mahasiswa mampu mengenal dan memahami Komunikasi kabel menggunakan UART, I2C dan SPI
- 1.2 Mahasiswa mampu mengimplementasikan program sederhana menggunakan Arduino Uno dan Sensor dengan Komunikasi kabel

2. ALAT DAN BAHAN PRAKTIKUM

- Arduino IDE
- Arduino Uno
- Kabel USB-B
- Kabel Jumper
- Project Board

3. DASAR TEORI

3.1 Komunikasi Kabel

Komunikasi kabel adalah cara bertukar data antar perangkat dengan menggunakan kabel sebagai penghubung fisik. Data dikirimkan dalam bentuk sinyal listrik atau cahaya yang merambat melalui kabel tersebut dari perangkat pengirim ke penerima. Jenis-jenis komunikasi kabel

3.2 UART

UART (Universal Asynchronous Receiver Transmitter) adalah Protokol komunikasi serial yang memungkinkan interaksi antara dua perangkat berbeda tanpa memerlukan sinyal clock yang terpisah.

Fungsi utama UART:

1. Transmitter (Pengirim): Menerima data secara bersamaan (paralel) dari prosesor, lalu mengubahnya menjadi data yang dikirimkan secara berurutan (serial), satu bit pada satu waktu.
2. Receiver (Penerima): Mengonversi aliran data serial yang diterima dari sumber eksternal kembali ke dalam format paralel untuk diproses oleh perangkat utama.

Kelebihan UART:

1. Sederhana dan Mudah diimplementasikan
2. Asinkron (Tidak membutuhkan Clock bersama)
3. Standar Universal
4. Dukungan Perangkat Lunak dan Komunitas yang Luas
5. Biaya Rendah

Kekurangan UART:

1. Kecepatan Data yang Terbatas
2. Tidak Efisien untuk Transfer Data dalam Volume Besar
3. Tidak mendukung multiple devices

3.3 I2C

I2C (Inter Integrated Circuit) adalah Protokol komunikasi serial yang memungkinkan satu perangkat utama (master) untuk terhubung dan mengendalikan beberapa perangkat tambahan (slave) dalam satu mode komunikasi.

Peran utama I2C:

1. Master adalah perangkat yang mengontrol komunikasi, dimana master menghasilkan sinyal clock (SCL) dan memulai serta mengakhiri proses transfer data. Contoh adalah Arduino dan ESP32.
2. Slave adalah perangkat yang dikontrol oleh Master, dimana slave merespons perintah dari master. Contoh adalah Sensor, memori dan lain-lain.

Kelebihan I2C:

1. Memiliki kemampuan Multi-Master dan Multi-Slave: Mendukung multiple master devices dalam satu bus
2. Hemat kabel: Hanya membutuhkan 2 kabel untuk menghubungkan banyak perangkat. Ini menghemat ruang, kompleksitas, dan biaya.
3. Mendukung kecepatan berbeda
 - ❖ Standard Mode: hingga 100 kbit/s
 - ❖ Fast Mode: hingga 400 kbit/s
 - ❖ High-Speed Mode: hingga 3.4 Mbit/s
4. Hemat Daya: Konsumsi daya relatif rendah dan dapat diimplementasikan dalam mode low-power

Kekurangan I2C:

1. Kecepatan terbatas: Lebih lambat dibandingkan SPI dan tidak cocok untuk aplikasi high-speed.
2. Jarak Terbatas: Hanya efektif untuk jarak pendek dan sensitif terhadap noise pada jarak panjang. Memerlukan pull-up resistor yang tepat untuk jarak tertentu
3. Overhead perangkat lunak: Implementasi di mikrokontroler seringkali membutuhkan lebih banyak pemrosesan perangkat lunak (software) dibandingkan SPI.

3.4 SPI

Serial Peripheral Interface (SPI) adalah Protokol komunikasi data serial sinkron yang memungkinkan mikrokontroler untuk bertukar data secara cepat dengan satu atau lebih perangkat pendukung dalam jarak dekat.

1. MISO (Master In Slave Out) adalah Saluran untuk mengirim data dari Slave ke Master.
2. MOSI (Master Out Slave in) adalah Saluran untuk mengirim data dari Master ke Slave.
3. SCLK (Serial Clock) adalah Sinyal clock yang dihasilkan oleh Master untuk menyinkronkan komunikasi.
4. SS/CS (Slave Select/Chip Select) adalah Sinyal untuk memilih Slave tertentu yang akan diajak komunikasi.

Kelebihan SPI:

1. Kecepatan sangat tinggi: SPI dapat berjalan pada kecepatan yang jauh lebih tinggi daripada I²C (bisa mencapai puluhan bahkan ratusan MHz).
2. Komunikasi Full-Duplex: Dapat mengirim dan menerima data secara bersamaan, yang meningkatkan throughput.
3. Protokol sederhana: Tidak ada alamat yang kompleks, tidak perlu pull-up resistor, dan overhead perangkat lunak lebih rendah

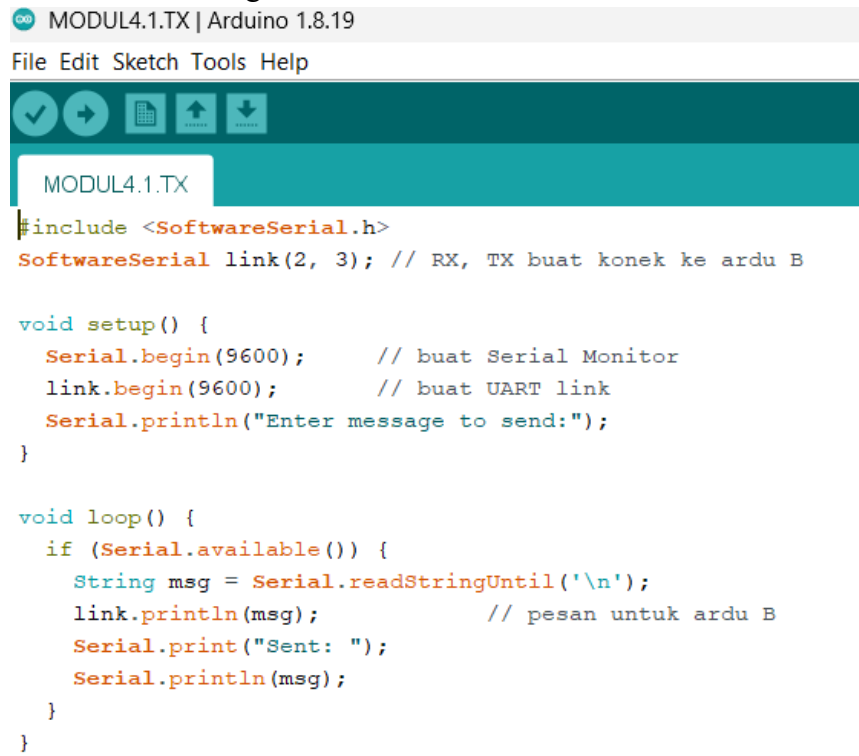
Kekurangan SPI:

1. Boros Pin: Setiap Slave tambahan membutuhkan satu pin SS/CS dedicated dari Master.
2. Jarak Komunikasi Pendek: SPI dirancang untuk komunikasi on-board dan tidak cocok untuk jarak jauh.

4. PERCOBAAN

4.1 Percobaan UART

1. Percobaan UART sebagai Tx



```
MODUL4.1.TX | Arduino 1.8.19
File Edit Sketch Tools Help

MODUL4.1.TX
#include <SoftwareSerial.h>
SoftwareSerial link(2, 3); // RX, TX buat konek ke ardu B

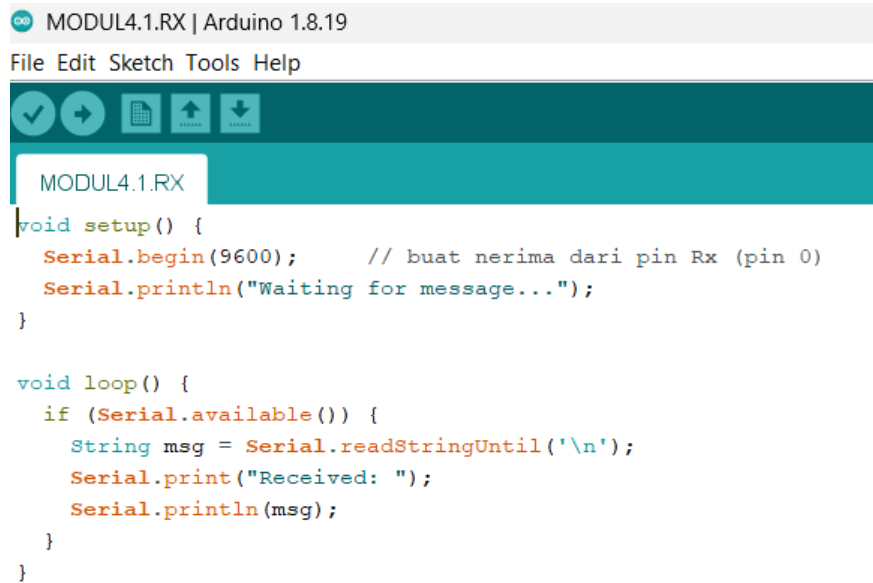
void setup() {
  Serial.begin(9600);      // buat Serial Monitor
  link.begin(9600);       // buat UART link
  Serial.println("Enter message to send:");
}

void loop() {
  if (Serial.available()) {
    String msg = Serial.readStringUntil('\n');
    link.println(msg);     // pesan untuk ardu B
    Serial.print("Sent: ");
    Serial.println(msg);
  }
}
```

Arduino Tx	Arduino Rx
Pin 3	Pin 0
GND	GND

Note: Upload terlebih dahulu program, baru dirangkai

2. Percobaan UART sebagai Rx



```
MODUL4.1.RX | Arduino 1.8.19
File Edit Sketch Tools Help
MODUL4.1.RX
void setup() {
  Serial.begin(9600);    // buat nerima dari pin Rx (pin 0)
  Serial.println("Waiting for message...");
}

void loop() {
  if (Serial.available()) {
    String msg = Serial.readStringUntil('\n');
    Serial.print("Received: ");
    Serial.println(msg);
  }
}
```

Arduino Rx	Arduino Tx
Pin 0	Pin 3
GND	GND

Note: Upload terlebih dahulu program, baru dirangkai

3. Percobaan UART Tx Rx secara bersamaan



```
MODUL4.1.TXRX | Arduino 1.8.19
File Edit Sketch Tools Help
MODUL4.1.TXRX
#include <SoftwareSerial.h>
SoftwareSerial link(2, 3); // RX, TX

void setup() {
  Serial.begin(9600); // buat Serial Monitor
  link.begin(9600); // buat UART link
  Serial.println("Type message and press Enter:");
}

void loop() {
  // If user types in Serial Monitor
  if (Serial.available()) {
    String msg = Serial.readStringUntil('\n');
    link.println(msg);
    Serial.print("You sent: ");
    Serial.println(msg);
  }

  // If a message arrives from the other Arduino
  if (link.available()) {
    String incoming = link.readStringUntil('\n');
    Serial.print("Received: ");
    Serial.println(incoming);
  }
}
```

Arduino 1	Arduino 2
Pin 2	Pin 3
Pin 3	Pin 2
GND	GND

Note: Upload terlebih dahulu program, baru dirangkai

4.2 Percobaan I2C

1. I2C sebagai Master

```

MODUL4.2.MASTER | Arduino 1.8.19
File Edit Sketch Tools Help
MODUL4.2.MASTER
#include <Wire.h>

void setup() {
  Wire.begin(); // mulai jadi I2C master
  Serial.begin(9600);
  Serial.println("format penulisan: <address>:<message>");
  Serial.println("contoh: 8:Hello or 9:Hi");
}

void loop() {
  if (Serial.available()) {
    String input = Serial.readStringUntil('\n');
    input.trim();

    int colonIndex = input.indexOf(':');
    if (colonIndex == -1) {
      Serial.println("format salah. Use: <address>:<message>");
      return;
    }

    int slaveAddr = input.substring(0, colonIndex).toInt();
    String message = input.substring(colonIndex + 1);
    message.trim();

    if (message.length() == 0) {
      Serial.println("tidak ada pesan terkirim!");
      return;
    }

    Wire.beginTransmission(slaveAddr);
    Wire.write(message.c_str());
    Wire.endTransmission();

    Serial.print("terkirim ke slave");
    Serial.print(slaveAddr);
    Serial.print(": ");
    Serial.println(message);
  }
}

```

Arduino Master	Arduino Slave 1	Arduino Slave 2
Pin A4	Pin A4	Pin A4
Pin A5	Pin A5	Pin A5
GND	GND	GND

Note: Upload terlebih dahulu program, baru dirangkai

2. I2C sebagai Slave

```

MODUL4.2.SLAVE | Arduino 1.8.19
File Edit Sketch Tools Help

MODUL4.2.SLAVE
#include <Wire.h>

void setup() {
  Wire.begin(8); // address I2C nya, angka 8-119 karena 0-7 dan 120-127 reserved (jgn dipakai)
  Wire.onReceive(receiveEvent);
  Serial.begin(9600);
  Serial.println("I2C Slave ready, menunggu data...");
}

void loop() {
  // diisi kode lain kalo dibutuhkan
  delay(100);
}

void receiveEvent(int bytes) {
  String msg = "";
  while (Wire.available()) {
    char c = Wire.read();
    msg += c;
  }
  Serial.print("Received: ");
  Serial.println(msg);
}
    
```

Arduino Master	Arduino Slave 1	Arduino Slave 2
Pin A4	Pin A4	Pin A4
Pin A5	Pin A5	Pin A5
GND	GND	GND

**Note: Upload terlebih dahulu program, baru dirangkai
Perbedaan Slave 1 dengan 2 pada address slave I2C-nya**

4.3 Percobaan SPI

1. SPI Master

```

MODUL4.3.MASTER | Arduino 1.8.19
File Edit Sketch Tools Help

MODUL4.3.MASTER
#include <SPI.h>

#define SS1 2 // Slave 1 select pin (pakai pin 2-9, jangan pakai 0,1,10)
#define SS2 3 // Slave 2 select pin

void setup() {
  Serial.begin(9600);
  SPI.begin();

  pinMode(SS1, OUTPUT);
  pinMode(SS2, OUTPUT);
  digitalWrite(SS1, HIGH);
  digitalWrite(SS2, HIGH);

  Serial.println("SPI Master ready.");
}

void loop() {
  if (Serial.available()) {
    String msg = Serial.readStringUntil('\n');
    msg.trim();

    int target = 0;
    
```

```

// pengecekan kirim ke slave yg mana
if (msg.startsWith("s1:")) {
    target = 1;
    msg = msg.substring(3); // remove "s1:"
}
else if (msg.startsWith("s2:")) {
    target = 2;
    msg = msg.substring(3); // remove "s2:"
}
else {
    Serial.println("format salah. Use s1: or s2:");
    return;
}

// Select target slave
if (target == 1) digitalWrite(SS1, LOW);
if (target == 2) digitalWrite(SS2, LOW);

// Send message byte-by-byte
for (int i = 0; i < msg.length(); i++) {
    SPI.transfer(msg[i]);
    delay(10);
}
SPI.transfer('\n');

// Deselect all
digitalWrite(SS1, HIGH);
digitalWrite(SS2, HIGH);

Serial.print("Sent to Slave ");
Serial.print(target);
Serial.print(": ");
Serial.println(msg);
}
}

```

Note: Upload terlebih dahulu program, baru dirangkai

Arduino Master	Arduino Slave 1	Arduino Slave 2
D11	D11	D11
D12	D12	D12
D13	D13	D13
D2	D10	-
D3	-	D10
GND	GND	GND

2. SPI Slave

```

MODUL4.3.SLAVE | Arduino 1.8.19
File Edit Sketch Tools Help
Save
MODUL4.3.SLAVE
#include <SPI.h>

volatile boolean messageComplete = false;
String receivedData = "";

void setup() {
  Serial.begin(9600);
  pinMode(MISO, OUTPUT);
  SPCR |= _BV(SPE);
  SPI.attachInterrupt();
  Serial.println("SPI Slave ready.");
}

ISR(SPI_STC_vect) {
  char c = SPDR;
  if (c == '\n') {
    messageComplete = true;
  } else {
    receivedData += c;
  }
}

void loop() {
  if (messageComplete) {
    Serial.print("Received: ");
    Serial.println(receivedData);
    receivedData = "";
    messageComplete = false;
  }
}

```

Arduino Master	Arduino Slave 1	Arduino Slave 2
D11	D11	D11
D12	D12	D12
D13	D13	D13
D2	D10	-
D3	-	D10
GND	GND	GND

Note: Upload terlebih dahulu program, baru dirangkai
Perbedaan Slave 1 dengan 2 pada slave select 1 dan 2 pada master

5. KESIMPULAN

UART, I2C, dan SPI adalah tiga protokol komunikasi kabel serial yang memiliki karakteristik dan aplikasinya masing-masing. UART merupakan protokol asinkron yang sederhana dan ideal untuk komunikasi point-to-point antara dua perangkat. Sementara itu, I2C lebih efisien dalam menghubungkan banyak perangkat (multi-slave) dengan hanya menggunakan dua kabel, meskipun kecepatannya terbatas. Di sisi lain, SPI menawarkan kecepatan transfer data yang sangat tinggi dan komunikasi full-duplex, namun membutuhkan lebih banyak kabel karena setiap perangkat slave memerlukan pin dedicated untuk seleksi.

6. LATIHAN

1. Apa yang dimaksud dengan Serial Clock (SCLK)?
2. Jelaskan perbedaan dari UART dan I2C menggunakan pemahaman sendiri!
3. Dalam program I2C sebagai slave, apa fungsi dari kode `Wire.begin(8)`?
4. Bagaimana master dalam SPI membedakan slavenya?

Terimakasih (★_★)