**Modul Praktikum Jaringan Komunikasi 06**

Protokol Pengiriman Data Synchronous berbasis Mikrokontroler

# Tujuan:

1. Mengenal dasar komunikasi data synchronous berbasis mikrokontroler
2. Mengenal I2C dan SPI pada mikrokontroler

# Kompetensi Dasar

## Inter Integrated Circuit

Inter Integrated Circuit biasa disebut sebagai I2C, Bus I2C atau IIC Bus. pada awalnya dikembangkan sebagai bus kontrol untuk menghubungkan mikrokontroler dan peripheral IC [1]. Inter-Integrated Circuit atau sering disebut sebagai komunikasi I2C adalah komunikasi yang dikembangkan oleh Philips SemiConductors, yang hanya menggunakan dua jalur komunikasi (2-wire) yaitu Synchronous Data (SDA) yang digunakan untuk mengirim dan menerima data (bi-directional) dan Synchronous Clock (SCL) yang digunakan untuk mengirim sinyal sinkronisasi. Protokol I2C untuk pengiriman satu byte data dapat dilihat pada Gambar 2.1.

A diagram of a number of data

Description automatically generated with medium confidence

Gambar 2.1. Dataframe Protokol Komunikasi I2C

Perangkat I2C menggunakan 2 buah pin open-drain dua arah dengan memberikan pull-up resistor untuk setiap garis bus sehingga berlaku seperti AND menggunakan kabel. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai Master dan Slave. Master adalah piranti yang memulai transfer data pada I2C Bus dengan membentuk sinyal Start, mengakhiri transfer data dengan membentuk sinyal Stop, dan membangkitkan sinyal clock. Slave adalah piranti yang dialamati Master.

Sinyal Start merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “1” menjadi “0” pada saat SCL “1”. Sinyal Stop merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “0” menjadi “1” pada saat SCL “1”. Kondisi sinyal Start dan sinyal Stop seperti tampak pada Gambar 2.2 [3].

A black rectangular object with lines

Description automatically generated with medium confidence

Gambar 2.2. Kondisi sinyal *start* dan *stop*

Sinyal dasar yang lain dalam I2C Bus adalah sinyal acknowledge yang disimbolkan dengan ACK Setelah transfer data oleh master berhasil diterima slave, slave akan menjawabnya dengan mengirim sinyal acknowledge, yaitu dengan membuat SDA menjadi “0” selama siklus clock ke 9. Ini menunjukkan bahwa Slave telah menerima 8 bit data dari Master. Kondisi sinyal acknowledge seperti tampak pada Gambar 2.3 [3].

A diagram of a line

Description automatically generated with medium confidence

Gambar 2.3. Sinyal ACK dan NACK pada I2C

## SPI (Serial Peripheral Interface)

Serial Peripheral Interface (SPI), protokol yang dibuat oleh Motorola pada tahun 1980-an, utamanya digunakan untuk komunikasi serial tersinkronisasi (synchronized serial communication) antara prosesor (master) dan IC yang dituju (slave). Empat buah jalur sinyal digunakan dalam komunikasi ini : Chip Select (CS), Serial Data Input (SDI), Serial Data Output (SDO), Serial Clock (SCKL). CS dan SCKL merupakan keluaran dari piranti master. Piranti slave menerima sinyal clock dan masukan chip select dari piranti master. Apabila sebuah piranti SPI tidak dipilih, jalur SDO dari piranti tersebut memiliki status high impedance state. Jumlah bit yang dikirim ke piranti slave bervariasi antar piranti. Setiap piranti slave memiliki sebuah internal shift register yang digunakan untuk pengiriman data.

Ada dua tipe koneksi antar piranti master dan slave. Pada koneksi tipe pertama, seluruh slave berbagi satu jalur CS. Jalur SDO dari piranti master dihubungkan ke jalur SDI dari piranti slave pertama. Jalur SDO dari piranti slave pertama dihubungkan ke jalur SDI piranti slave kedua. Begitu seterusnya hingga hubungan terakhir ke jalur SDI piranti master. Pada koneksi tipe kedua, setiap jalur SDI piranti slave terhubung pada jalur SDO piranti master. Setiap piranti slave memiliki jalur CS masing-masing. Setiap jalur SDO piranti slave terhubung ke jalur SDI piranti master.

Jenis komunikasi ini juga mendukung mode multiple master. Laju serial clock mempunyai jangkauan dari 30 kHz hingga 3 MHz, bergantung pada piranti yang digunakan. Beberapa literature menyebutkan CS sebagai SS, SDI sebagai MOSI, dan SDO sebagai MISO.

# Langkah Percobaan

## I2C

1. Buat rangkaian seperti pada blok diagram di bawah ini dengan menghubungkan SDA dan SCL pada *device master* dan *device slave*.

A diagram of a diagram of a computer

Description automatically generated

Gambar 3.1. Blok diagram protokol komunikasi I2C

### Percobaan 1: I2C pada Arduino

1. Buka Arduino IDE
2. Buat program untuk pengiriman data via protokol komunikasi I2C seperti



Gambar 3.2. Contoh program I2C *master* pada Arduino



Gambar 3.3. Contoh program I2C *slave* pada Arduino

1. Siapkan 2 buah arduino. Arduino ke-1 sebagai I2C *master* dan Arduino ke-2 sebagai I2C *slave*.
2. *Verify* dan *build* program yang telah dibuat pada masing-masing Arduino 1 dan 2
3. Hubungkan pin Tx Arduino *master* pada osiloskop
4. Variasikan karakter, clock, dan address I2C sesuai tabel percobaan
5. Amati dataframe hasil variasi pada osiloskop dan serial monitor Arduino *slave*

### Percobaan 2: I2C pada PSoC

* + - 1. Buka perangkat lunak PSoC Creator
      2. Klik tab File > New > Project
      3. Susun komponen blok digital pada PSoC Project sesuai Gambar 3.4 untuk *master* dan Gambar 3.5 untuk *slave*.

A screenshot of a computer

Description automatically generated

Gambar 3.4. Komponen blok yang digunakan pada PSoC *Master*

A screenshot of a computer

Description automatically generated

Gambar 3.5. Komponen blok yang digunakan pada PSoC *Slave*

* + - 1. Atur blok I2C master seperti pada Gambar 3.6.

A screenshot of a computer program

Description automatically generated

Gambar 3.6. Konfigurasi komponen I2C *master*

* + - 1. Atur komponen I2C *slave* seperti pada Gambar 3.7

A screenshot of a computer program

Description automatically generated

Gambar 3.7. Konfigurasi komponen I2C *slave*

* + - 1. Buat kode program untuk I2C *master* seperti pada Gambar 3.8.

A computer code with text

Description automatically generated with medium confidence

Gambar 3.8. Contoh kode program PSoC *master*

* + - 1. Buat kode program I2C *slave* seperti pada Gambar 3.9.

A computer code with text

Description automatically generated with medium confidence

Gambar 3.9. Contoh kode program I2C *slave*

* + - 1. Clean and build program masing-masing project ke masing-masing PSoC sesuai perannya masing-masing
      2. Hubungkan Tx pada PSoC *master* dengan probe osiloskop
      3. Variasikan karakter, address, dan buffer size pada bagian main.c *master* sesuai tabel percobaan
      4. Variasikan frekuensi clock pada PSoC master dan slave sesuai dengan tabel percobaan.
      5. Buka Arduino IDE
      6. Pilih port yang tersambung dengan PSoC *Slave* lalu buka Serial Monitor
      7. Amati dan catat hasil yang diperoleh pada serial monitor

## SPI (Serial Peripheral Interface)

* Buat rangkaian pada Arduino dan PSoC seperti pada .. dengan menghubungkan SCLK, MISO, MOSI, dan SS pada Arduino dan PSoC *Master* dan *Slave*



Gambar 3.10. Blok diagram protokol komunikasi SPI

### Percobaan 3: SPI pada Arduino

1. Buka Arduino IDE
2. Buat kode program Arduino *master* seperti pada Gambar 3.11.



Gambar 3.11. Contoh kode program Arduino untuk SPI *master*

1. Buat kode program untuk Arduino sebagai SPI *slave*, seperti pada Gambar 3.12.



Gambar 3.12. Contoh kode program Arduino SPI *slave*

1. Siapkan 2 buah Arduino. Arduino 1 sebagai SPI *master* dan Arduino 2 sebagai SPI *slave*.
2. Verify dan build kode program yang telah dibuat pada masing-masing Arduino
3. Variasikan karakter, clock, mode master, dan mode *slave* sesuai dengan tabel percobaan
4. Buka Arduino IDE dan sambungkan dengan Arduino *Slave*
5. Pilih port yang sesuai dan serial monitor pada Arduino IDE
6. Amati data yang diterima Arduino *slave* pada serial monitornya.

### Percobaan 4: SPI pada PSoC

Buka PSoC Creator

Buat 2 (dua) project pada PSoC, dengan nama SPIMaster dan SPISlave

Di dalam project SPIMaster, pada bagian TopDesign.cysch masukkan blok komponen SPI master dari Component Catalog.

A screenshot of a computer

Description automatically generated

Gambar 3.13. Komponen SPI Master pada Component Catalog PSoC

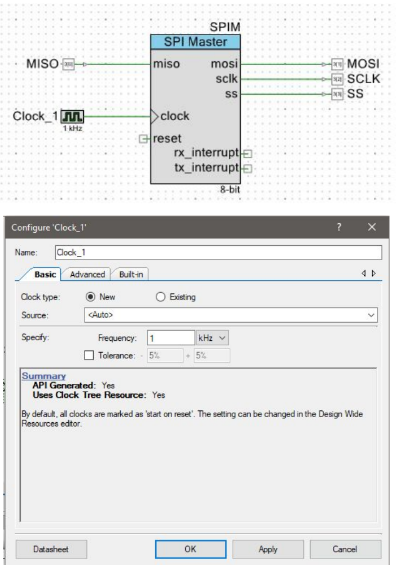
Atur komponen SPI Master, gunakan pengaturan komponen sesuai Gambar 3.14. Pemberian nama komponen akan mempengaruhi nama pada program.

A screenshot of a computer

Description automatically generated

Gambar 3.14. Pengaturan komponen SPI *Master*

Tambahkan pin digital input pada MISO, digital output pada MOSI, SCLK, dan SS. Logic Low pada reset dan Clock sebesar 1kHz pada clock dan pada bagian Tolerance seperti pada Gambar 3.15.



Gambar 3.15. Pengaturan penamaan pin dan clock pada SPI *master*

Buat kode program SPI *master* seperti pada Gambar 3.16.

A screenshot of a computer program

Description automatically generated

Gambar 3.16. Contoh kode program PSoC SPI *master*

Konfigurasi pin / ports untuk *input* dan *output* SPI *master* sesuai dengan Gambar 3.17.

A screenshot of a computer

Description automatically generated

Gambar 3.17. Nomor pin *input* dan *output* pada SPI *master*

Clean dan build *program* lalu unggah program ke PSoC pertama yang berlaku sebagai SPI *master*.

Dalam project SPISlave, pada workspace TopDesign.cysch masukkan komponen SPI slave yang didapatkan dari Component Catalog seperti pada Gambar 3.18.

A screenshot of a computer

Description automatically generated

Gambar 3.18. Komponen SPI Slave pada Component Catalog PSoC

Atur SPI Slave pada komponen, gunakan pengaturan komponen seperti pada Gambar 3.19. Pemberian nama komponen akan mempengaruhi nama pada program.

A screenshot of a computer

Description automatically generated

Gambar 3.19. Pengaturan komponen SPI *Slave*

Tambahkan pin digital output pada MISO, digital output MOSI, SCLK, dan SS, lalu logic Low pada reset dan Clock sebesar 1 kHz. Pada bagian clock, bagian tolerance di-nonaktifkan hingga sesuai dengan Gambar 3.20. Tambahkan komponen UART dan port keluaran Tx.

A screenshot of a computer

Description automatically generated

Gambar 3.20. Komponen SPI Slave dan UART

Buat kode program SPI *Slave* pada main.c untuk Project SPISlave seperti pada Gambar 3.21.

A computer screen shot of a program code

Description automatically generated

Gambar 3.21. Kode program SPI *Slave*

* + - 1. Konfigurasikan pin input-output SPI Slave seperti pada Gambar 3.22.

A screenshot of a computer

Description automatically generated

Gambar 3.22. Konfigurasi pin input-output SPI *Slave*

* + - 1. *Clean* dan *Build*program lalu unggah program ke PSoC kedua yang berperan sebagai SPI *Slave*.
      2. Hubungkan dengan kabel port MOSI master ke MOSI Slave, MISO Master ke MISO Slave, dan berlaku seterusnya untuk SS, SCLK dan Ground
      3. Buka Arduino IDE pada laptop yang terhubung dengan PSoC *Slave* dan pilih port yang sesuai.
      4. Pilih serial monitor pada Arduino IDE dan lihat data yang diterima oleh *slave*.

# Tabel Percobaan

## Percobaan 1: I2C pada Arduino

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mikrokontroler | Data yang dikirim | Clock Master | Clock Slave | Address Master | Adddress Slave | Data di Serial Monitor |
| Arduino | Lab. SSTK | 100 kHz | | 8 | 8 |  |
| Punya | 400 kHz | | 8 | 25 |  |
| Lambang | 1 MHz | | 21 | 19 |  |
| Yaitu | 100 kHz | 400 kHz | 65 | 65 |  |
| Badak | 100 kHz | 1 MHz | 27 | 63 |  |
| Kesimpulan |  | | | | | |

Percobaan 2: I2C pada PSoC

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Kata | Clock Master dan Slave | Clock pada Slave | Address pada Master | Address pada Slave | Buffer Master | Buffer Slave | Data di Serial Monitor |
| Jarkom Hore | 400kHz | | 8 | 8 | 1 | 1 |  |
| Ada di | 1Mhz | | 13 | 76 | 10 | 2 |  |
| Lab. SSTK | 100kHz | | 16 | 16 | 12 | 12 |  |
| Punya | 433kHz | 100kHz | 17 | 19 | 3 | 7 |  |
| Lambang | 143kHz | 189kHz | 36 | 36 | 12 | 8 |  |
| Yaitu | 673kHz | 673kHz | 33 | 33 | 6 | 3 |  |
| Badak | 127kHz | 127kHz | 26 | 26 | 50 | 50 |  |
| Kesimpulan | |  | | | | | |

Percobaan 3: SPI pada Arduino

|  |  |  |
| --- | --- | --- |
| Pesan yang dikirim | Clock Divider Master | Data yang diterima |
| Tahun ini | 2 |  |
| Jarkom | 4 |  |
| Asyik | 8 |  |
| Ada | 16 |  |
| Responsinya | 32 |  |
| Kesimpulan |  | |

Percobaan 4: SPI pada PSoC

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Karakter | Clock di Master | Clock di Slave | Mode Master | Mode Slave | Data di Serial Monitor |
| L | 10kHz | | CPHA = 0  CPOL = 0 | CPHA = 0  CPOL = 0 |  |
| A | CPHA = 0  CPOL = 0 | CPHA = 0  CPOL = 1 |  |
| B | CPHA = 0  CPOL = 0 | CPHA = 1  CPOL = 0 |  |
| . | CPHA = 0  CPOL = 0 | CPHA = 1  CPOL = 1 |  |
| S | CPHA = 0  CPOL = 1 | CPHA = 0  CPOL = 0 |  |
| S | CPHA = 0  CPOL = 1 | CPHA = 0  CPOL = 1 |  |
| T | CPHA = 0  CPOL = 1 | CPHA = 1  CPOL = 0 |  |
| K | CPHA = 0  CPOL = 1 | CPHA = 1  CPOL = 1 |  |
| A | CPHA = 1  CPOL = 0 | CPHA = 0  CPOL = 0 |  |
| D | CPHA = 1  CPOL = 0 | CPHA = 1  CPOL = 0 |  |
| A | CPHA = 1  CPOL = 0 | CPHA = 1  CPOL = 1 |  |
| B | CPHA = 1  CPOL = 1 | CPHA = 1  CPOL = 1 |  |
| A | 1Mhz | 3Mhz | CPHA = 0  CPOL = 0 | |  |
| D | 1Mhz | 400kHz | CPHA = 0  CPOL = 0 | |  |
| A | 3Mhz | 1Mhz | CPHA = 1  CPOL = 1 | |  |
| K | 10kHz | 100khz | CPHA = 1  CPOL = 1 | |  |
| Kesimpulan | |  | | | |