

Malang, 2022

(.....)

MODUL IV

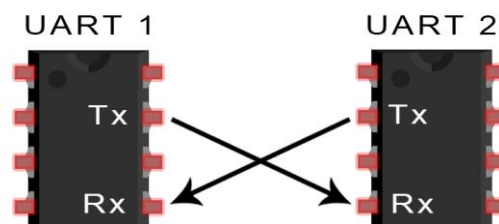
KOMUNIKASI DIGITAL UART, SPI, I2C

4.1 Pendahuluan

Komunikasi digital adalah teknologi yang berbasis sinyal elektrik komputer, sinyalnya bersifat terputus-putus dan menggunakan sistem bilangan biner. Bilangan biner akan membentuk kode digital (1 dan 0 berdasarkan adanya arus listrik atau tidak, yang diatur oleh transistor).

4.1.1 Komunikasi digital UART (Universal Asynchronous Receiver Transmitter)

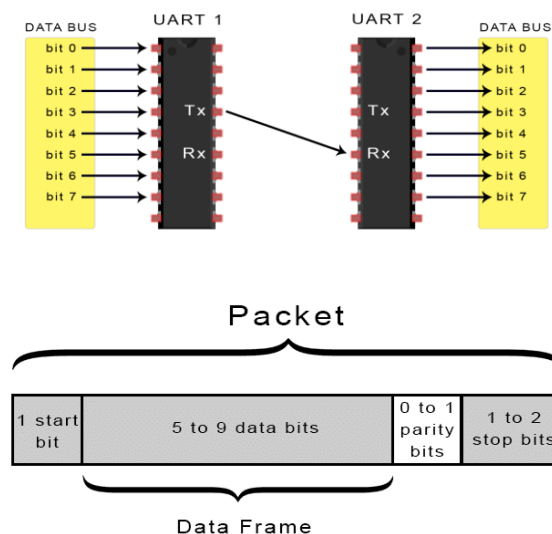
Dalam komunikasi UART, dua UART berkomunikasi secara langsung satu sama lain. UART pengirim mengubah data paralel dari perangkat pengontrol seperti CPU ke dalam bentuk serial, mengirimkannya secara serial ke UART penerima, yang kemudian mengubah data serial kembali menjadi data paralel untuk perangkat penerima. Ketika UART dihubungkan dengan PC, kita membutuhkan hardware berupa USB(Universal Serial Bus) to TTL(Transistor Transistor Logic), karena PC menggunakan komunikasi USB, jadi ketika Arduino berkomunikasi dengan PC kita harus menambahkan hardware berupa USB to TTL.



Wires Used	2
Maximum Speed	Any speed up to 115200 baud, usually 9600 baud
Synchronous or Asynchronous?	Asynchronous
Serial or Parallel?	Serial
Max # of Masters	1
Max # of Slaves	1

➤ Cara kerja UART

Data ditransfer dari bus data ke UART pengirim dalam bentuk paralel. Setelah UART yang mentransmisikan mendapatkan data paralel dari bus data, ia menambahkan bit awal, bit paritas, dan bit stop, membuat paket data. Selanjutnya, paket data dikeluarkan secara serial, kemudian UART penerima mengubah data kembali ke bentuk paralel dan menghapus bit awal, bit paritas, dan bit stop. Terakhir UART penerima mentransfer paket data secara paralel ke bus data di sisi penerima. Referensi lebih lengkap [disini](#)

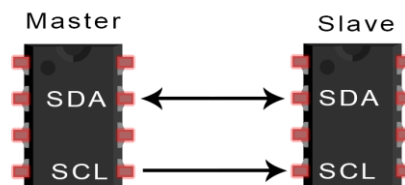


- Start bits : UART transmisi merubah jalur transmisi data dari tegangan HIGH ke LOW selama satu siklus clock. Ketika UART penerima mendeteksi transisi tegangan tinggi ke rendah, ia mulai membaca bit dalam baud rate yang ditentukan.
- Data frame : berisi data aktual yang sedang ditransfer. Panjangnya bisa 5 bit hingga 8 bit (jika bit paritas digunakan). Jika tidak ada bit paritas yang digunakan, data frame dapat memiliki panjang 9 bit.

- Parity bits(optional) : Setelah UART penerima membaca data frame, ia menghitung jumlah bit yang bernilai 1 dan memeriksa apakah totalnya genap atau ganjil. Jika bit paritas adalah 0 (paritas genap), bit yang bernilai 1 dalam data frame harus berjumlah genap. Jika bit paritas adalah 1 (paritas ganjil), bit yang bernilai 1 dalam data frame harus berjumlah ganjil. Tetapi jika bit paritas adalah 0, dan totalnya ganjil, UART mengetahui bahwa bit dalam bingkai data telah berubah.
- Stop bits : pengirim merubah jalur transmisi data dari tegangan LOW ke tegangan HIGH untuk setidaknya dua durasi bit.

4.1.2 Komunikasi digital I2C (Inter Integrated Circuit)

I2C menggabungkan fitur terbaik dari SPI dan UART. Dengan I2C, Anda dapat menghubungkan beberapa slave ke satu master (seperti SPI) dan Anda dapat memiliki beberapa master yang mengendalikan satu, atau beberapa slave. Ini sangat berguna ketika Anda ingin memiliki lebih dari satu mikrokontroler yang mencatat data ke satu kartu memori atau menampilkan teks ke satu LCD. Referensi lebih lengkap [disini](#)



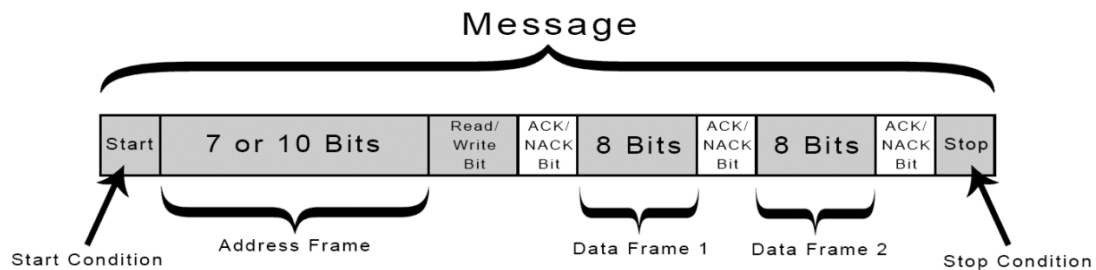
Keterangan :

SDA (Serial Data), SCL (Serial Clock).

Wires Used	2
Maximum Speed	Standard mode= 100 kbps
	Fast mode= 400 kbps
	High speed mode= 3.4 Mbps
	Ultra fast mode= 5 Mbps
Synchronous or Asynchronous?	Synchronous
Serial or Parallel?	Serial
Max # of Masters	Unlimited
Max # of Slaves	1008

➤ Cara kerja I2C

Data ditransfer dalam pesan, pesan dipecah menjadi frame data. Setiap pesan memiliki frame alamat dari slave dalam bentuk biner, dan satu atau lebih frame data yang berisi data yang sedang dikirim atau dibaca.



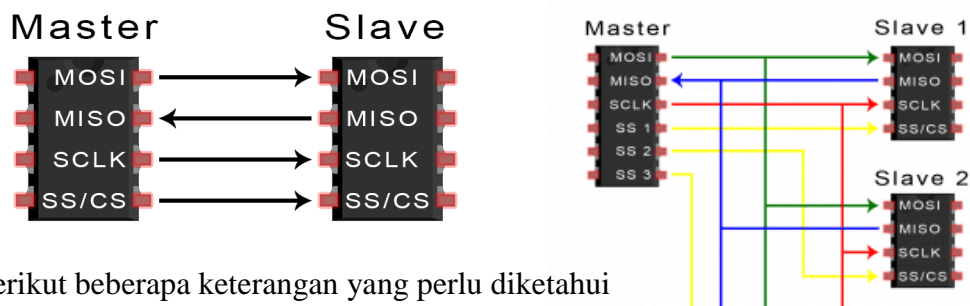
- Start condition : jalur SDA beralih dari HIGH ke LOW, sebelum jalur SCL beralih dari HIGH ke LOW.
- Stop condition : jalur SDA beralih dari LOW ke HIGH, setelah jalur SCL beralih dari LOW ke HIGH.
- Address Frame : urutan 7 atau 10 bit untuk setiap slave, yang berfungsi untuk mengidentifikasi slave ketika master ingin berkomunikasi dengannya.
- Read / Write bit : menentukan apakah master mengirimkan data ke slave(level bit LOW) atau meminta data dari slave (level bit HIGH).
- ACK / NACK bit : setiap frame dalam sebuah pesan diikuti bit acknowledge / no-acknowledge, jika frame alamat data berhasil diterima, bit ACK dikembalikan ke master dari device slave. Jika data benar, bit ACK mengembalikan satu bit LOW.

4.1.3 Komunikasi digital SPI (Serial Peripheral Interface)

SPI adalah protokol komunikasi umum yang digunakan oleh banyak perangkat berbeda. Misalnya, modul pembaca kartu SD , modul pembaca kartu RFID , dan pemancar/penerima nirkabel 2,4 GHz semuanya menggunakan SPI untuk berkomunikasi dengan mikrokontroler.

Salah satu manfaat unik dari SPI adalah kenyataan bahwa data dapat ditransfer tanpa gangguan. Sejumlah bit dapat dikirim atau diterima dalam aliran yang berkelanjutan. Dengan I2C dan UART, data dikirim dalam paket, terbatas pada jumlah bit tertentu. Kondisi mulai dan berhenti menentukan awal dan akhir setiap paket, sehingga data terganggu selama transmisi. Referensi lebih lengkap klik [disini](#)

Konfigurasi paling sederhana dari SPI adalah master tunggal, sistem budak tunggal, tetapi satu master dapat mengontrol lebih dari satu.



Berikut beberapa keterangan yang perlu diketahui

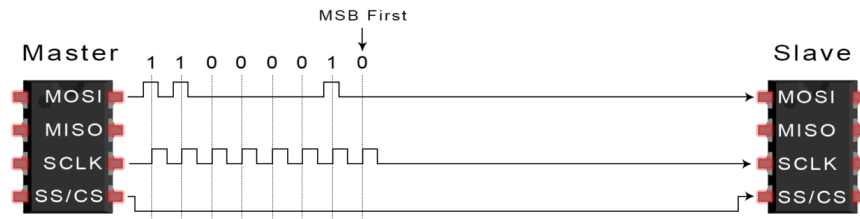
- MOSI (Master Output/Slave Input)
- MISO (Master Input/Slave Output)
- SCLK (Clock)
- SS/CS (Slave Select/Chip Select)

Wires Used	4
Maximum Speed	Up to 10 Mbps
Synchronous or Asynchronous?	Synchronous
Serial or Parallel?	Serial
Max # of Masters	1
Max # of Slaves	Theoretically unlimited*

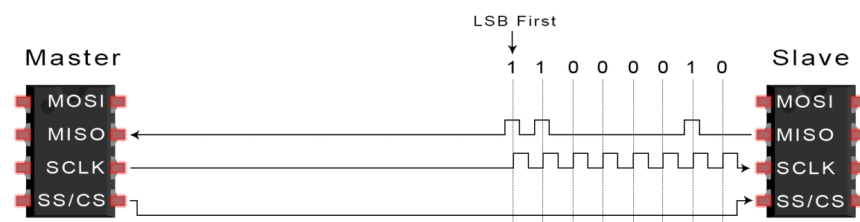
➤ Cara kerja SPI

Sinyal clock menyinkronkan output bit data dari master ke pengambilan sampel bit oleh slave. Satu bit data ditransfer dalam setiap siklus clock, sehingga kecepatan transfer data ditentukan oleh frekuensi sinyal clock. Komunikasi SPI selalu dimulai oleh master karena master mengonfigurasi dan menghasilkan sinyal clock. Master dapat memilih slave mana yang ingin diajak bicara dengan mengatur jalur CS/SS slave ke level tegangan rendah. Dalam keadaan idle, non-transmisi, saluran pilih slave disimpan pada level tegangan tinggi. Beberapa pin

CS/SS mungkin tersedia pada master, yang memungkinkan beberapa slave disambungkan secara paralel.



Data yang dikirimkan dari master ke slave biasanya dikirim dengan most signifikan bit terlebih dahulu.



Data yang dikirim dari slave kembali ke master biasanya dikirim dengan bit paling tidak signifikan terlebih dahulu.

4.2 Tujuan

Setelah praktikum selesai dilakukan peserta diharapkan dapat :

- Memahami prinsip kerja dari komunikasi UART, I2C, dan SPI
- Menganalisa program yang ada di masing-masing percobaan.
- Melihat perbedaan dari 3 komunikasi di atas.
- Memahami pin i/o yang bisa digunakan untuk berkomunikasi.
- Melakukan wiring antar perangkat.

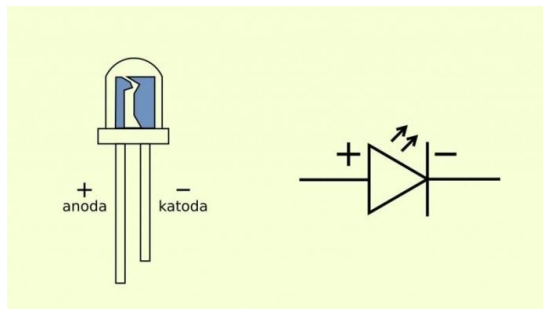
4.3 Bahan Percobaan

Bahan yang digunakan pada percobaan ini :

1. Arduino uno 2 buah.



2. LED 4 buah.



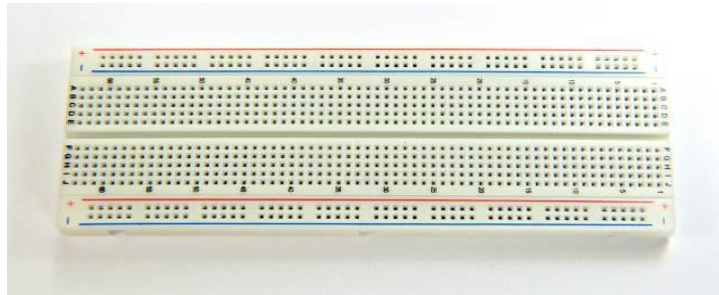
3. Resistor 330/220 4 buah



4. Arduino USB Cable 2.0.



5. Projectboard .

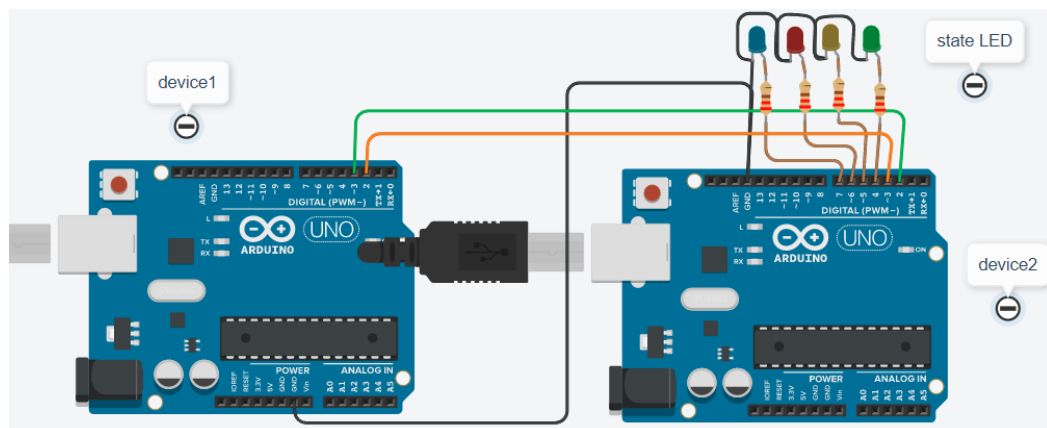


6. Kabel Jumper.



4.4 Percobaan I komunikasi UART

4.4.1 rangkaian komunikasi UART

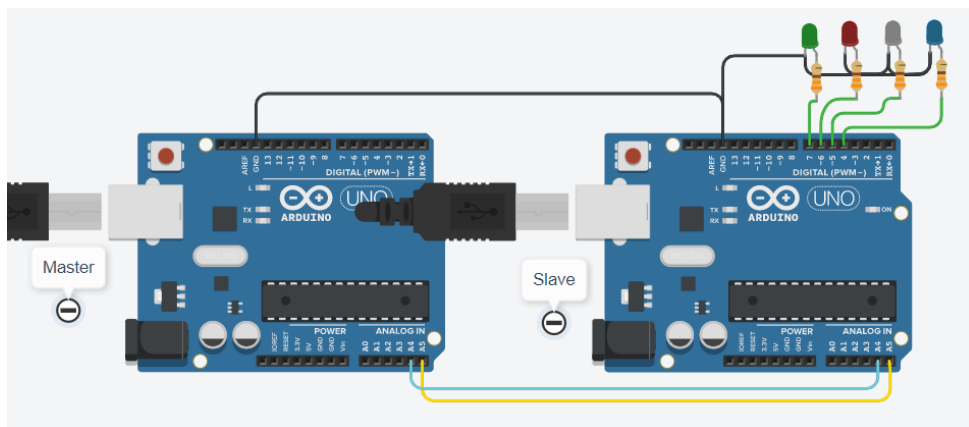


4.4.2 langkah-langkah komunikasi UART

- Hubungkan semua perangkat sesuai dengan rangkaian di atas.
- Download program [disini](#)
- Masukkan program Device1 dan Device2 pada board Arduino.
- Hidupkan kedua ardino dengan port usb 2.0.
- Buka serial monitor untuk board device1r pada software Arduino ide.
- Masukkan angka satuan pada serial monitor antara 1 sampai 4(bisa angka satuan yang lain).
- Tekan send.
- Lihat pesan-pesan yang ada di serial monitor.

4.5 Percobaan II komunikasi I2C

4.5.1 rangkaian komunikasi I2C



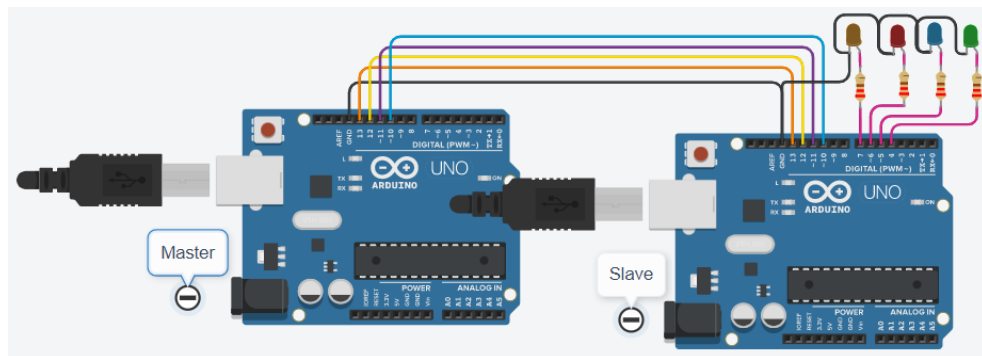
4.5.2 langkah-langkah komunikasi I2C

- Hubungkan semua perangkat sesuai dengan rangkaian di atas.
- Download program [disini](#)
- Upload program Master dan Slave pada board Arduino.
- Hidupkan kedua ardino dengan port usb 2.0.

- Buka serial monitor untuk board Master pada software Arduino ide.
- Masukkan angka satuan pada serial monitor antara 1 sampai 4(bisa angka satuan yang lain).
- Tekan send.
- Lihat pesan-pesan yang ada di serial monitor.

4.6 Percobaan III komunikasi SPI

4.6.1 rangkaian komunikasi SPI



4.6.2 langkah-langkah komunikasi SPI

- Hubungkan semua perangkat sesuai dengan rangkaian di atas.
- Download program [disini](#)
- Upload program Master dan Slave pada board Arduino.
- Hidupkan kedua ardino dengan port usb 2.0.
- Buka serial monitor untuk board Master pada software Arduino ide.
- Masukkan angka satuan pada serial monitor antara 1 sampai 4(bisa angka satuan yang lain).
- Tekan send.
- Lihat pesan-pesan yang ada di serial monitor.

4.7 Memprogram dan analisa komunikasi UART

4.7.1 Buat program komunikasi UART



4.7.2 Analisa program UART

.....

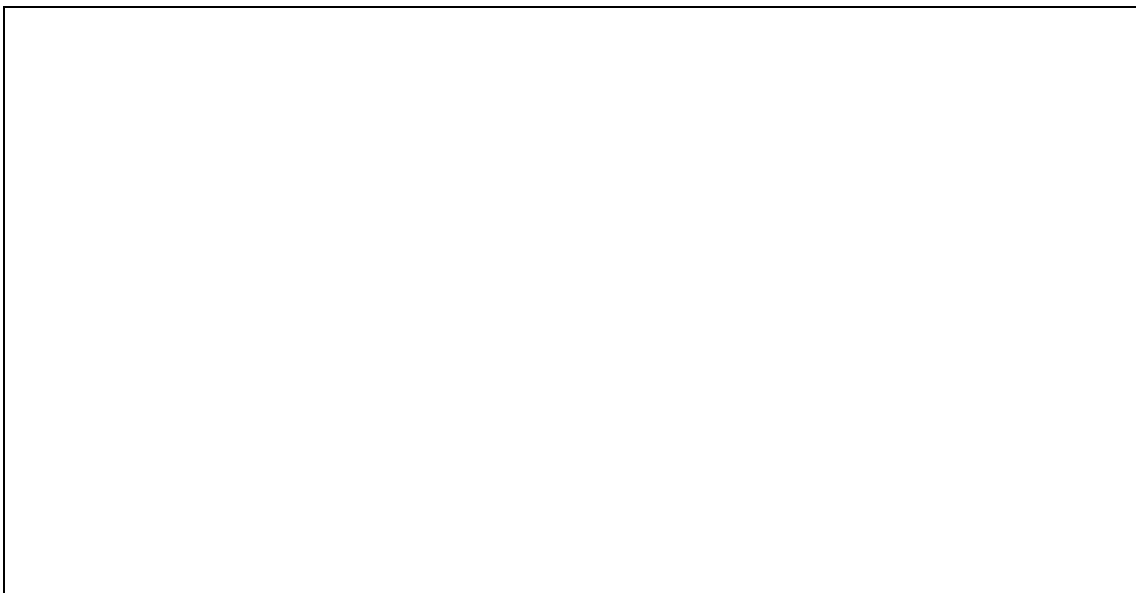
.....

.....

.....

.....

4.7.3 Capture Percobaan UART



4.7.4 Analisa percobaan UART

.....

.....

.....

.....

.....

.....

4.8 Memprogram dan analisa komunikasi I2C

4.8.1 Buat program komunikasi I2C

4.8.2 Analisa program I2C

.....

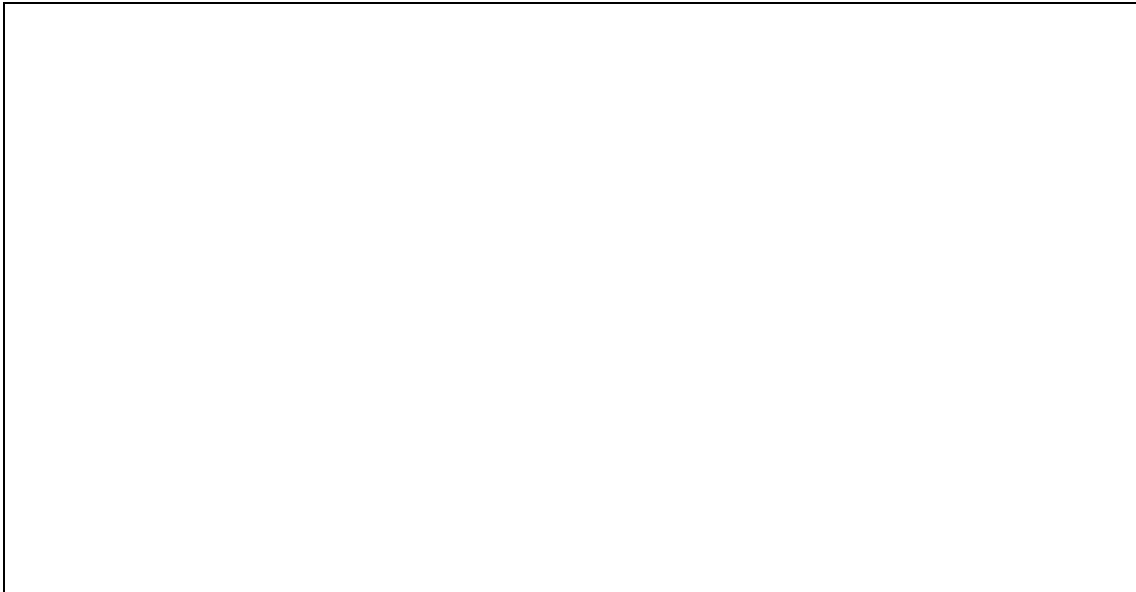
.....

.....

.....

.....

4.8.3 Capture Percobaan I2C



4.8.4 Analisa Percobaan I2C

.....

.....

.....

.....

.....

4.9 Memprogram dan analisa komunikasi SPI

4.9.1 Buat program komunikasi SPI



4.9.2 Analisa program SPI

.....

.....

.....

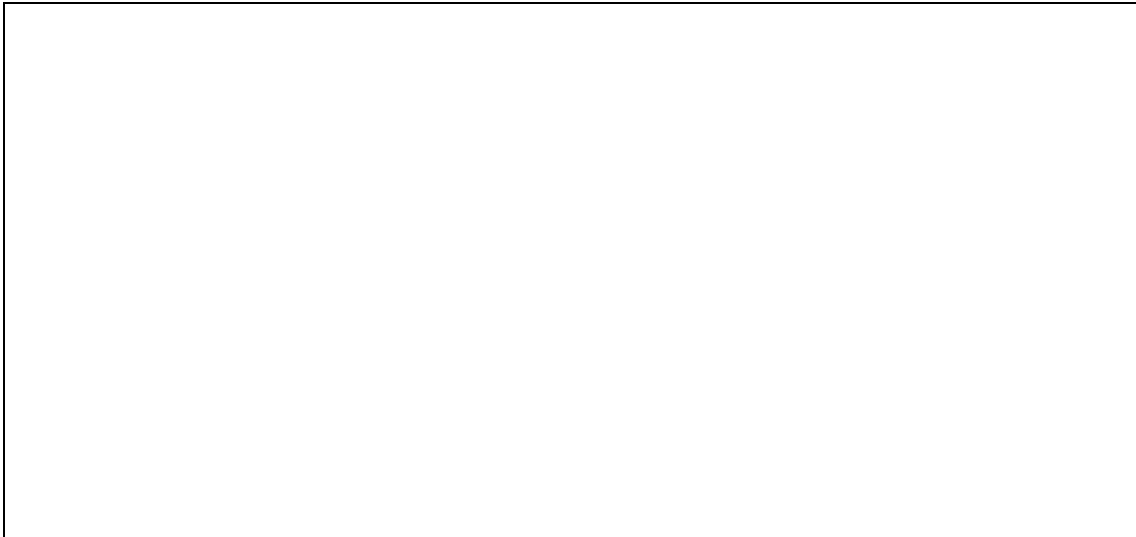
.....

.....

.....

.....

4.9.3 Capture komunikasi SPI



4.9.4 Analisa Percobaan SPI

.....

.....

.....

.....

.....

.....

.....

4.10 Tugas

1. Jelaskan secara teori bagaimana cara komunikasi SPI dan I2C melakukan pengalamatan terhadap Slave.

Jawab :

.....

.....

.....

.....

2. Apakah komunikasi UART bisa berkomunikasi lebih dari 2 device ? Jelaskan.

Jawab :

.....

.....

.....

.....

3. Kenapa komunikasi SPI dan I2C menggunakan nama Master dan Slave, sedangkan UART tidak, jelaskan menurut pemahaman anda masing-masing!

Jawab :

.....

.....

.....

.....

4.11 Kesimpulan

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....