**SISTEM TERDISTRIBUSI**

1. **Tujuan**

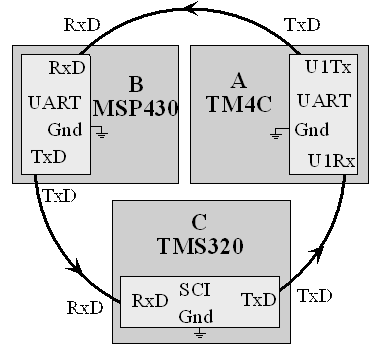
* Memahami sinkronisasi I/O
* Memahami model peranti I/O (*busy*, *done*, *off*)
* Menguasai pemrograman UART
* Membangun sistem terdistribusi
* Menguasai konversi antara bilangan dan ASCII

1. **Peralatan yang digunakan**

* Keil μVision v5
* 2 Tiva C Series LaunchPad
* Stellaris® ICDI Drivers
* Kabel UTP yang sangat panjang
* Osiloskop

1. **Landasan teori**

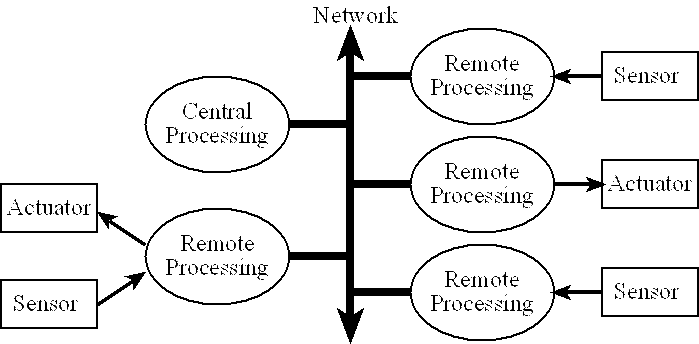
Jaringan adalah koleksi antarmuka yang berbagi medium fisis dan protokol data. Jaringan memungkinkan *task* perangkat lunak dalam sebuah komputer untuk berkomunikasi dan sinkronisasi dengan *task* perangkat lunak yang berjalan pada komputer lain. Untuk sistem *embedded*, jaringan menyediakan sarana untuk komputasi terdistribusi. Topologi jaringan menentukan bagaimana komponen saling terhubung. Contoh topologi meliputi cincin, *bus* dan multi-hop. Gambar 10 memperlihatkan jaringan cincin tiga mikrokontroler.



Gambar 10. Jaringan cincin sederhana dengan tiga *node*, terhubung menggunakan *port* serial

Kelebihan jaringan cincin di atas adalah murah dan dapat diterapkan pada setiap mikrokontroler dengan *port* serial. Perhatikan bahwa mikrokontroler tidak harus tipe atau laju yang sama.

Ketika berhadapan dengan masalah kompleks, bisa dikembangkan solusi pada satu sistem komputer yang canggih dan terpusat. Alternatif lain, solusi terdistribusi dapat digunakan menggunakan banyak komputer terhubung oleh jaringan. Elemen pengolahan dalam Gambar 11 dapat berupa komputer yang canggih, mikrokontroler, *application-specific integrated circuit* (ASIC), atau sensor/aktuator cerdas.



Gambar 11. Pengolahan terdistribusi menempatkan *input*, *output* dan pengolahan pada banyak lokasi terhubung bersama dengan jaringan

1. **Eksperimen**

Kembangkan jaringan komunikasi antara dua Tiva LaunchPad. Saklar adalah *input*, LED adalah *output*, dan UART digunakan untuk komunikasi. Ada lima pertanyaan dan tiga jawaban. Informasi dikodekan sebagai warna pada LED.

|  |  |  |
| --- | --- | --- |
| Warna LED | Pertanyaan | Jawaban |
| Merah | Apakah kamu di kantor? |  |
| Kuning | Apakah kamu bahagia? |  |
| Hijau | Apakah kamu lapar, ingin makan siang? |  |
| Biru | Apakah kamu haus, ingin bertemu untuk minum? |  |
| Biru muda | Haruskah aku datang ke kantormu untuk berbicara? |  |
| Putih |  | Ya |
| Ungu |  | Mungkin |
| Tidak berwarna |  | Tidak |

Operator memilih pesan yang dikirim dengan penekanan saklar SW1. Ketika pemilihan pesan, LED menampilkan pesan yang akan dikirim. Setiap kali operator menekan saklar SW1, sistem akan siklus melalui 8 kemungkinan warna pada LED. Ketika operator menekan SW2, pesan terkirim. Isi pesan dikodekan sebagai karakter ASCII ‘0’ sampai ‘7’ (0x30 sampai 0x37) dan dikirim melalui UART. *Driver* dari Program 8 digunakan.

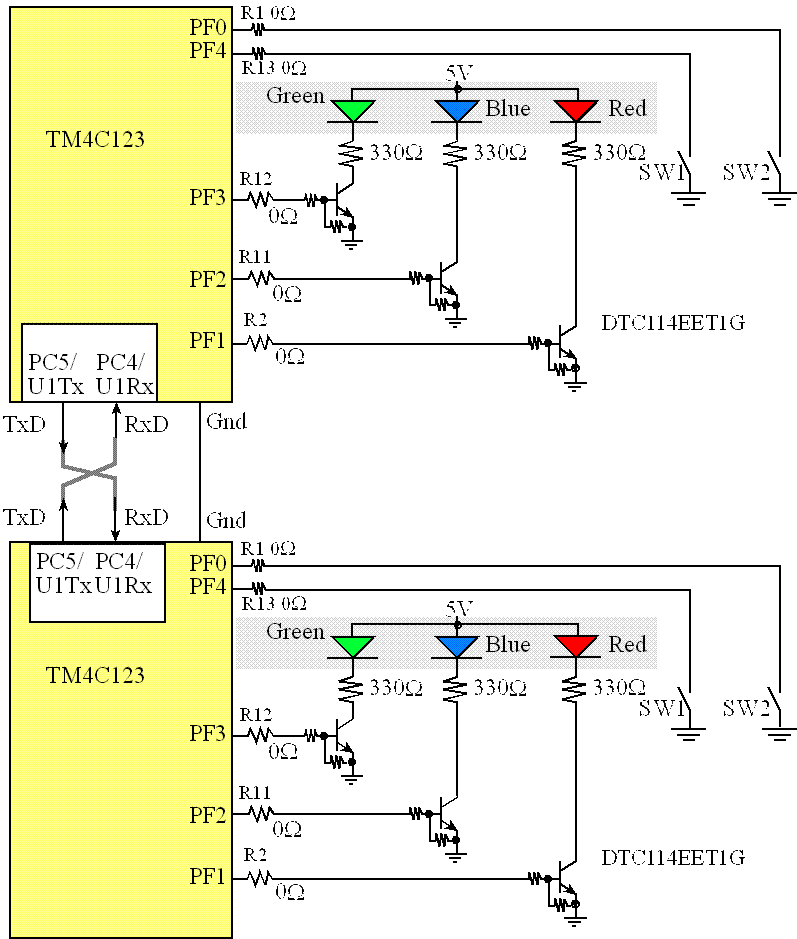
|  |
| --- |
| **//  red, yellow, green, light blue, blue, purple,  white,  dark**  **const long ColorWheel[8] = {0x02,0x0A,0x08,0x0C,0x04,0x06,0x0E,0x00};**  **int main(void){ unsigned long SW1,SW2;**  **long prevSW1 = 0;        // previous value of SW1**  **long prevSW2 = 0;        // previous value of SW2**  **unsigned char inColor;   // color value from other microcontroller**  **unsigned char color = 0; // this microcontroller's color value**  **PLL\_Init();              // set system clock to 80 MHz**  **SysTick\_Init();          // initialize SysTick**  **UART\_Init();             // initialize UART**  **PortF\_Init();            // initialize buttons and LEDs on Port F**  **while(1){**  **SW1 = GPIO\_PORTF\_DATA\_R&0x10; // Read SW1**  **if((SW1 == 0) && prevSW1){    // falling of SW1?**  **color = (color+1)&0x07;     // step to next color**  **}**  **prevSW1 = SW1; // current value of SW1**  **SW2 = GPIO\_PORTF\_DATA\_R&0x01; // Read SW2**  **if((SW2 == 0) && prevSW2){    // falling of SW2?**  **UART\_OutChar(color+0x30);   // send color as '0' - '7'**  **}**  **prevSW2 = SW2; // current value of SW2**  **inColor = UART\_InCharNonBlocking();**  **if(inColor){ // new data have come in from the UART??**  **color = inColor&0x07;     // update this computer's color**  **}**  **GPIO\_PORTF\_DATA\_R = ColorWheel[color];  // update LEDs**  **SysTick\_Wait10ms(2);        // debounce switch**  **}**  **}** |

Program 8. Jaringan komunikasi tingkat tinggi (C11\_Network)

Ketika *frame* UART diterima, data dikodekan dan data ‘0’ sampai ‘7’ ditampilkan sebagai warna pada LED. Gambar 12 memperlihatkan perangkat keras, yang menggunakan 3 kabel terhubung ke setiap Tiva LaunchPad.

**Instruksi:**

* Hubungkan dua Tiva LaunchPad berdasarkan Gambar 12.
* Jalankan C:\Keil\TExaSware\C11\_Network\Network.uvproj pada masing-masing Tiva LaunchPad.
* Periksa kesesuaian Program 8 dengan Network.uvproj.
* Hubungkan kanal 1 osiloskop ke PC5 Tiva yang pertama.
* Hubungkan kanal 2 osiloskop ke PC4 Tiva yang pertama.
* Tekan saklar SW1 pada Tiva yang pertama sehingga LED berwarna merah.
* Tekan saklar SW2 pada Tiva yang pertama.
* Amati apa yang terlihat pada osiloskop.
* Tekan saklar SW1 pada Tiva yang kedua sehingga LED berwarna putih.
* Tekan saklar SW2 pada Tiva yang kedua.
* Amati apa yang terlihat pada osiloskop.
* Bermainlah dengan semua warna yang ada pada masing-masing Tiva LaunchPad, sambil mengamati tampilan pada osiloskop.



Gambar 12. Sistem terdistribusi menggunakan dua Tiva LaunchPad terhubung bersama oleh UART

1. **Bibliografi**

<http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C11_SerialInterface.htm>