ESKISEHIR TECHNICAL UNIVERSITY

DEPARTMENT OF

ELECTRICAL AND ELECTRONICS ENGINEERING

EEM 449 – EMBEDDED SYSTEM DESIGN

PROJECT 15

Fall, 2020-2021

Name/Surname: FATMA ÖZÜDOĞRU

ID: 42613176000

Date: 24.01.2021

1.Project Description

In this project, a mobile robot platform was controlled over the internet. Project is worked on Tiva C Series EK-TM4C1294XL and an integrated development environment (IDE) is Code Composer Studio which version is 10.0. The mobile robot has two DC motors and these motors are controlled by a L298N Voltage Regulator Dual Motor Driver. On launchpad, it is created TCP/IP server that waits for incoming connections over the port 5030. The server waits for commands from the Internet Server. Client can send HELLO, GETTIME, LEFT X(Left motor running time in seconds), RIGHT Y(Right motor running time in seconds), EXEC and QUIT.

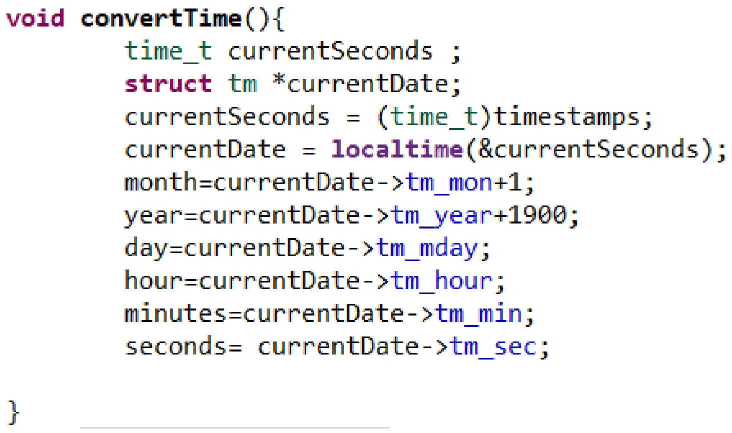
Once program receives “LEFT 2” command, it will understand that LEFT motor will be active for 2 seconds. Similarly, “RIGHT 3” means that RGHT motor will be active for 3 seconds. When program receives “EXEC” commands, it will start executing the previous commands. All synchronization is provided using events and mailboxes. Also, this program receives time data from the NTP server once and updates the received time with the timer in the system. When the client sends “GETTIME” to server, server sends data that includes year, month, day, hour, minutes and seconds to client.

2.Project Steps

2.1. Getting Time Data from NTP Server

This step of the project took place in the following order.

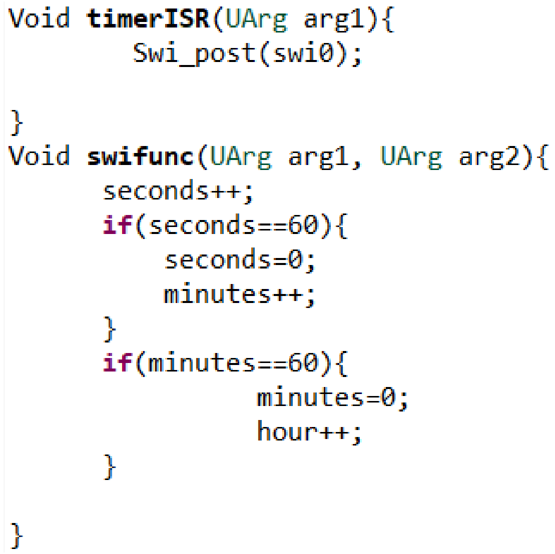
- TCP / IP Client socket task is created for connecting NTP server.
- Program is connected NTP Server with server IP(132.163.96.2) address and server port(37). Server port and IP adddres is defined global variable.
- 4 byte time data is got and is converted it to timestamp.
- Timestamp is converted to date(year, month, day, hour, minutes, seconds) with the convertTime function(Figure 1) in the program.



```
void convertTime(){
        time_t currentSeconds ;
        struct tm *currentDate;
        currentSeconds = (time_t)timestamps;
        currentDate = localtime(&currentSeconds);
        month=currentDate->tm_mon+1;
        year=currentDate->tm_year+1900;
        day=currentDate->tm_mday;
        hour=currentDate->tm_hour;
        minutes=currentDate->tm_min;
        seconds= currentDate->tm_sec;

}
```

Figure 1 *convertTime Function*

- The converted day, month, year, hour, minutes and seconds data is assigned to global variables.
- It is setup a Timer HWI called every second to update the current time. Time HWI posts a SWI in order to update clock values (year, month, day, hour, minutes, seconds) in every second.



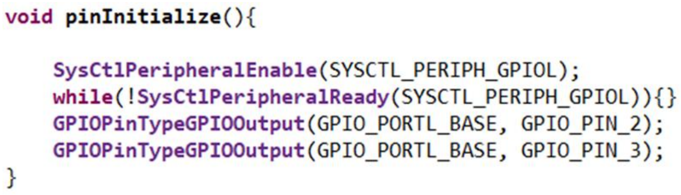
```
Void timerISR(UArg arg1){
        Swi_post(swi0);

}
Void swifunc(UArg arg1, UArg arg2){
      seconds++;
      if(seconds==60){
          seconds=0;
          minutes++;
      }
      if(minutes==60){
                minutes=0;
                hour++;
      }

}
```

Figure 2 *HWI and SWI Function*

2.2. Controlling of DC motors

This step of the project took place in the following order.

- GPIO pins are selected to run the motors on the card (PL2, PL3). These pins are configured as digital output with pinInitialize function(Figure 3).



```
void pinInitialize(){

    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOL);
    while(!SysCtlPeripheralReady(SYSCTL_PERIPH_GPIOL)){}
    GPIOPinTypeGPIOOutput(GPIO_PORTL_BASE, GPIO_PIN_2);
    GPIOPinTypeGPIOOutput(GPIO_PORTL_BASE, GPIO_PIN_3);
}
```

Figure 3 *pinInitialize Function*

- There are ENA(Left motor channel activation pin), IN1& IN2 (Left motor inputs), IN3& IN4 (Right motor inputs), ENB(Right motor channel activation pin), OUT1&OUT2(Left motor output), OUT3&OUT4 (Right motor output),VCC(Supply voltage input (4.8V-24V)),GND(Ground connection) and 5V(5V output) pins on the L298N Voltage Regulator Dual Motor Driver. Left motor is connected to OUT1&OUT2, right motor is connected to OUT3&OUT4, VCC and GND pins are connected to 3S Lipo battery. PL2 is connected to IN3 pin, PL3 is connected to IN1 pin. Ground pin of the board and the ground pin of motor driver are connected to each other.

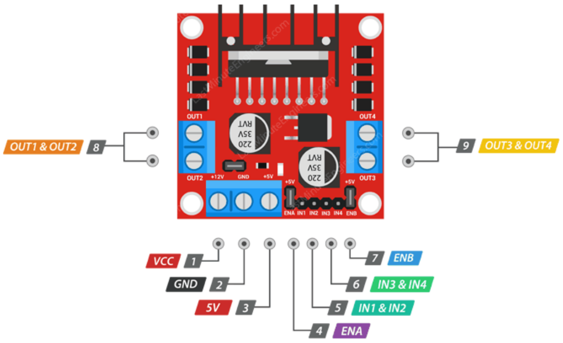


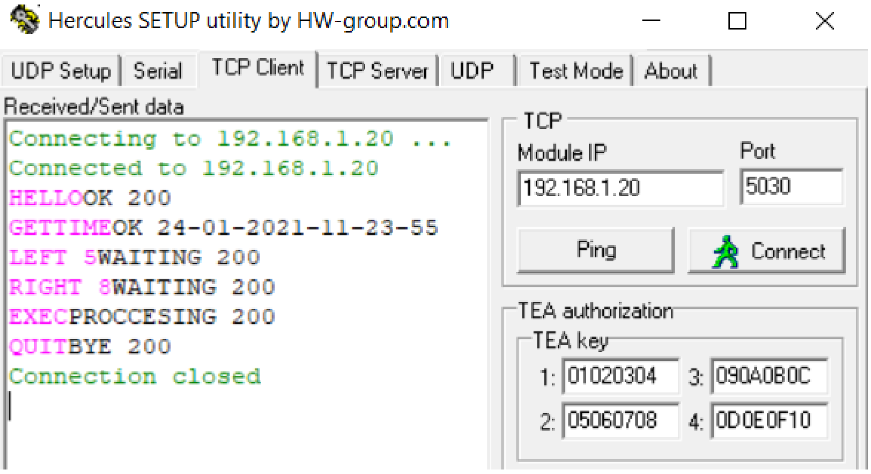
Figure 4 *L298N Voltage Regulator Dual Motor Driver Pins*

- Motor is worked by GPIOPinWrite (GPIO_PORTL_BASE, GPIO_PIN_3, GPIO_PIN_3) function and stopped by GPIOPinWrite (GPIO_PORTL_BASE, GPIO_PIN_3,0) function.

2.3. Creating a TCP / IP Server

This step of the project took place in the following order.

- TCP / IP Server socket task is created and TCP / IP Server waits for incoming connections over the port 5030.
- TCP/IP Server waits for data to come over internet.Data is sent to the created server over the internet using the tcp client over the Hercules program.
- Client can send HELLO, GETTIME, LEFT X(Left motor running time in seconds), RIGHT Y(Right motor running time in seconds), EXEC and QUIT. The server sends a response to the customer as in Figure 5, according to the incoming data.



Received/Sent data

```
Connecting to 192.168.1.20 ...
Connected to 192.168.1.20
HELLOOK 200
GETTIMEOK 24-01-2021-11-23-55
LEFT 5WAITING 200
RIGHT 8WAITING 200
EXECPROCCESING 200
QUITBYE 200
Connection closed
```

TCP

Module IP: 192.168.1.20

Port: 5030

TEA authorization

TEA key

1: 01020304
3: 090A0B0C
2: 05060708
4: 0D0E0F10

Figure 5 *Response of Server*

- The processing and synchronization of data received from the server is provided by mailbox and event. The information on which motor will run and how many seconds it will run was posted to task0function from serverSocketTask with mailbox. Data is arranged to using placement() function in the task0function. When the information for the right and left motor is received, the task0Function is posted to the task1Function with event. When the data that is “EXEC” is came to the serverSocketTask, serverSocketTask posts to the task1Function with event. When event_id_00 (motor information) and event_id_01 (EXEC) comes to the task1Function, the motors are run according to the information given.