

KARABÜK ÜNİNERSİTESİ

Mühendislik FAKÜLTESİ

Mekatronik Mühendisliği BÜLÖMÜ

4.SINIF 2.ÖĞRETİM

Mekatronik Proje Uygulaması I

**Dr.Öğr.Üyesi Can Bülent FİDAN**

**ARAÇ TAKİP SİSTEMİ**

ABDULJAWAD KAYALI

2018-2019

İçindekiler

1. Tanıtım ve amaç
2. TM4C123GH6PM
3. GPS cipi ve NMEA
4. SIM800L ve AT-COMMAND
5. GOOGLE MAPS
6. Keil proramı
7. Sonuç ve gerçekleştirilmeyen hususlar
8. Referanslar

Tanıtım ve amaç

GPS (Gloal Position System) sistemin sağladığı konum belirleme özelliği baz alarak Araç takip systemin projesi gerçekleştirmeye çalışıldı .

NEO-6M-0-001 çipin sağladığı konum bilgisi mekrodenetleyici aktarmak suretiyle TM4C123H6PM mikroişlemci ortamına UART haberleşme mödülünü kullanarak okunudu ve SIM800L çipi ile internet’e bağlanarak konum bilgileri bir web sayfasında veya android ugylaması üzerinde gösterilmeye planlanıldı .

Bu proje pıyasada mevcuttur , araba kiralık şirketleri ve nakliye şerketleri sıklıkla kullanmaktadırlar .

Aşağdaki şekilde proje anlatımaya çalışılmıştır

|  |
| --- |
|  |

Şekil-1

TM4C123GH6PM

Arm şirketin sunduğu bu mikroişlemci Cortex-M4 ailesindendir .

Tercih etme sebepim bu mikroişlemci’de sunulan haberleşme mödülün zenginliğinden kaynaklanmaktadır , ayrıca bu proje ile birlikte yeni bir mikroişlemcinin çalışması öğrenmek istediğimden tercih ettim .

Genel özellikler :

* 32-bitlik ARM Cortex M4 80 Mhz CPU
* 256 kb flash / 32 kb RAM / 2kb EEPROM
* 12-bit SAR ADC (1MSPS) , comparators, Timers, DMA
* 2 CAN bus mödülü
* 8 UART mödülü
* SPI/ I2C mödülleri

TM4C123GH6PM hakkında bu raporda bazı bilgiler anlatılacaktır .

Data sheetini inceledikten sonra bu mikroişlemcinin çalışma prensebiözellikleri ve kullanımını öğrendim .

Bir mikroişlemci’de bulunan registerleri öğrenmeden işlem yaptırmak zordur , TM4C123GH6PM ‘de 15 register bulunmaktadır . R1- R12 arasında bu 12 register ( GENERAL PRUPOSE REGİSTER ) , bunları programımız’da istediğimiz işlemler’de kullanabiliriz.

Özel kullanıma sahip olan registerler ise : R13 registeri ( stack pointer) , R14 register ( link register ) , R15 register ( program counter ).

Bootloader :bilgisayar’den yükletilecek olan kod mikroişlemciye aktarmak için bir araça ihtayç duyulmaktadır , bazı mikroişlemcilerde bunu ana mikroişlemciyi kullanarak gerçekleştirmektedirler fakat Tiva çipin’de ana mikroişlemcinin yanında başka bir TM4C123GH6PM mikroişlemci kullanılmaktadır .

JTAG (4 bitlik bağlantı) kullanarak ana mikroişlemci ile yükletmede kullanılacak mikroişlemci arasındaki bağlantıyı sağlanmaktadır .

Tiva C Series TM4C123GH6PM schematic diyagramı datasheetinde paylaşılmaktadır fakat PCB tasarım programlarında kütüphanesi üreten şirket veya program geliştircilerin tarafında ücersiz olarak paylaşılmamaktadır ,internet’te araştırdıktan sonra PCB Artist program kullanarak Jonathan Valvano tarafından tasarımda gerek duyulacak elmanlar tasarlandı .

Bu kütüphaneye kullanarak proje için istenilen PCB tasarımı yapıldı , proje’de iki UART’a ihtiyaç var , kullancının mikroişlemciye müdahele etmemesi için bootloader kullanılmadı .

GPS ve SİM800L pcb tasarımı yapılacaktı fakat bu çiplerde sinyal filtrelerinin hasasyeti göz önünde bulundurarak hazır olarak üretilen çipleri kullanılmaya karar verildi .

Aşağdaki resimler’de PCB’nin tasarm şikelleri bulunmaktadır.

|  |
| --- |
|  |

Şekil-2

|  |
| --- |
|  |

Şekil-3

GPS cipi ve NMEA

Arabanın konumunu belirleyen algılayacı GPS mödülüdür , proje’de kullanıllan mödül NEO-6M-0-001 bu mödülün Teknik özellikleri :

* Power supply range 3V-5V
* Ceramic anten
* EEPROM for saving the configuration data when powered off backup battary
* LED signal indictor
* Mounting hole diameter :3mm

Default baud rate :9600 bps

GPS’ın çıkışları bağlanması :

* VCC 3.3V/5V güç kaynağına bağlanır
* GND : GND bağlantısı
* TxD : mikroişlemci RxD girişine
* RxD : mikroişlemcinin TxD girişine

GPS çıkışlarından ( Rx ve Tx ) elde edilen bilgileri mikroişlemcinin UART bağlantıları sayısında mikroişlemci’de okunur .

Mikroişlemci’de okunan GPS bilgileri NMEA Sentence olarak adlandılır .

NMEA Sentence :

Cümlelerden ibarettir her cümlenin başındaki bulunan kilemeler bu cümlede bulunan bilgileri tanıtmak ve hangi anlama geldikleri açıklamak içindir .

NMEA ‘yı daha iyi birr şikelde anlamak için NMNEA sitesindeki bulunan açıklamaları ve anlatımları okumak gerkir .

Bu sitede bulunan bu bilgilerden bazıları burada örnek vermek amacıyla paylaşılacatır

Site’de bulunduğu gibi ingilizce olarak paylaşılacaktır .

Daha detaylı bilgiler bu siteye ziyaret ederek elde edilebilir

* Çipi ulaşan sinyal durumu için :

$GPMSS,55,27,318.0,100,\*66

where:

55 signal strength in dB

27 signal to noise ratio in dB

318.0 Beacon Frequency in KHz

100 Beacon bitrate in bps

\*66 checksum

**\*** **GGA** - essential fix data which provide 3D location and accuracy data.

$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,\*47

Where:

GGA Global Positioning System Fix Data

123519 Fix taken at 12:35:19 UTC

4807.038,N Latitude 48 deg 07.038' N

01131.000,E Longitude 11 deg 31.000' E

1 Fix quality: 0 = invalid

1 = GPS fix (SPS)

2 = DGPS fix

3 = PPS fix

4 = Real Time Kinematic

5 = Float RTK

6 = estimated (dead reckoning) (2.3 feature)

7 = Manual input mode

8 = Simulation mode

08 Number of satellites being tracked

0.9 Horizontal dilution of position

545.4,M Altitude, Meters, above mean sea level

46.9,M Height of geoid (mean sea level) above WGS84

ellipsoid

(empty field) time in seconds since last DGPS update

(empty field) DGPS station ID number

\*47 the checksum data, always begins with \*

\* $GPGSA,A,3,04,05,,09,12,,,24,,,,,2.5,1.3,2.1\*39

Where:

GSA Satellite status

A Auto selection of 2D or 3D fix (M = manual)

3 3D fix - values include: 1 = no fix

2 = 2D fix

3 = 3D fix

04,05... PRNs of satellites used for fix (space for 12)

2.5 PDOP (dilution of precision)

1.3 Horizontal dilution of precision (HDOP)

2.1 Vertical dilution of precision (VDOP)

\*39 the checksum data, always begins with \*

\* **GLL** - Geographic Latitude and Longitude is a holdover from Loran data and some old units may not send the time and data active information if they are emulating Loran data. If a gps is emulating Loran data they may use the LC Loran prefix instead of GP.

$GPGLL,4916.45,N,12311.12,W,225444,A,\*1D

Where:

GLL Geographic position, Latitude and Longitude

4916.46,N Latitude 49 deg. 16.45 min. North

12311.12,W Longitude 123 deg. 11.12 min. West

225444 Fix taken at 22:54:44 UTC

A Data Active or V (void)

\*iD checksum data

$GPGSV,2,1,08,01,40,083,46,02,17,308,41,12,07,344,39,14,22,228,45\*75

Where:

GSV Satellites in view

2 Number of sentences for full data

1 sentence 1 of 2

08 Number of satellites in view

01 Satellite PRN number

40 Elevation, degrees

083 Azimuth, degrees

46 SNR - higher is better

for up to 4 satellites per sentence

\*75 the checksum data, always begins with \*

**RMC** - NMEA has its own version of essential gps pvt (position, velocity, time) data. It is called RMC, The Recommended Minimum, which will look similar to:

$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W\*6A

Where:

RMC Recommended Minimum sentence C

123519 Fix taken at 12:35:19 UTC

A Status A=active or V=Void.

4807.038,N Latitude 48 deg 07.038' N

01131.000,E Longitude 11 deg 31.000' E

022.4 Speed over the ground in knots

084.4 Track angle in degrees True

230394 Date - 23rd of March 1994

003.1,W Magnetic Variation

\*6A The checksum data, always begins with \*

Kullandığım NEO-6M-0-001 GPS mödülü NMEA versiyonu 1.5 ve elde edilecek olan NMEA’ler genel olarak aşağda gösterildiği gibi :

$GPRMC,162254.00,A,3723.02837,N,12159.39853,W,0.820,188.36,110706,,,A\*74

$GPVTG,188.36,T,,M,0.820,N,1.519,K,A\*3F

$GPGGA,162254.00,3723.02837,N,12159.39853,W,1,03,2.36,525.6,M,-25.6,M,,\*65

$GPGSA,A,2,25,01,22,,,,,,,,,,2.56,2.36,1.00\*02

$GPGSV,4,1,14,25,15,175,30,14,80,041,,19,38,259,14,01,52,223,18\*76

$GPGSV,4,2,14,18,16,079,,11,19,312,,14,80,041,,21,04,135,25\*7D

$GPGSV,4,3,14,15,27,134,18,03,25,222,,22,51,057,16,09,07,036,\*79

$GPGSV,4,4,14,07,01,181,,15,25,135,\*76

$GPGLL,3723.02837,N,12159.39853,W,162254.00,A,A\*7C

$GPZDA,162254.00,11,07,2006,00,00\*63

Yukarda anlatıllan parametreleden faydalanarak her cümlenin ne anlama geldiğini bulabiliriz burada bulunmayan parametreler <https://www.gpsinformation.org/dale/nmea.html> sitesinde elde edilebilir .

Konum bilgisine aşağdaki paramtre’den alınabilir

$GPRMC,162254.00,A,3723.02837,N,12159.39853,W,0.820,188.36,110706,,,A\*74

Fakat bu bilgiye google mapsa aktarlıdığı zaman başka bir konum yani çipin bulunmadığı konumuy gösterecektir sepebi :

4807.038,N Latitude 48 deg 07.038' N

01131.000,E Longitude 11 deg 31.000' E

Bu iki parametre’de bulunan zaman bilgisini adlandırılan 07.038 ve 31.000 sayılarını 60’a bölerek ve sırasıyla 4807.038,N ve 01131.000,E ekleyerek 3-5m gibi konum bilgiler elde edilir .

GPS çipini kullanırken dikkat edilmesi gereken hususlar :

1. Açık bir yerde veya pencereye yakın kullanılmadığı zamanlar’da veriler elde edilmez
2. Mikroişlemce’de bulunan 5V çıkışına bağlanabilir fakat Sım800l mödülü içinb tasarlanacak olan 4V lük güç kaynağı kullanılabilirsiniz
3. Elde eddilen bilgiler karekter olarak üretilir

Aşağdaki şekilde mikroişlemci ile GPS mödülün bağlanmış halde gösterilmektedir

|  |
| --- |
|  |

Şekil-4

SIM800L ve AT-COMMAND

SIM800L mödülü kullanarak sim kartı ve mikroişlemci arasında bağlantı kuruldu ve haberleşmeye sağladıktan sonra SIM800L çipine AT-COMMAND komutları göndirelerek internet’e bağlanması sağlandı .

SIM800L özellikleri:

* Dört bantlı 850/900/1800/1900 Mhz
* GPRS çok slotlu sınıfı 12/10
* Mobil istasyonu B sınıfı
* Besleme voltajı 3.4-4.4 V olaarak data sheetinde gözükmektedir fakat amper değeri 2.1 A’e anlık ve hızlı olarak bazı durumlarda çıkabilir , bu yüzden 6.2V -2A sağlayabilen bir güç kaynağı kullanıldı

Mödülün şebekye bağlanma esnasında veya data alış-veriş yapıldığında amperin değişimleri maruz kaldığı fark edildi , çipimiz 3.4-4.4 V arasında şebekye bağlandıktan sonra çalışabildiği fark edildi.

SIM800L sağladığı hizmetlerden birisi HTTP ‘den faydalandı , ve bu bağlantıyı kullanarak web sayfalarında Get metodu kullanarak bilgiler okundu , fakat bir server sahip olmadığım’dan dolay ücretsiz hizmet veren web sayfalarında faydalanmaya çalıştım ancak POST metodunu kullanımama izin verilmedi bu yüzden gps çipinden gelen bilgileri internet ortamına , hazırladığım web sayfasına aktaramadım , projem’de gerçekleştirmediğim kısım buydu .

AT-COMMAND kullanarak SIM800L çipine komutlar gönderilebilir ,bu komutlara SIM800L cevap üretmektedir ve her gönderilen komut bize mödülün durumu aktarabilir veya gönderdiğimiz komuta göre mödülün ayarları yapabiliriz .

Aşağda bu komutların bazı örnekler ve anlamları gösterilmektedir

* AT+CGACT? : çipin şebekye bağlanmasından emin olmak için
* AT+CGQREQ? : Quality of serivce
* AT+CBST? : Bearer service configuration
* AT+CR? : service reporting

Ve bu komutları gibi bir çok komut bulunmaktadır, tüm komutların hangi anlama geldiğini SIMCOM verdiği datasheetinde bulunmaktadır

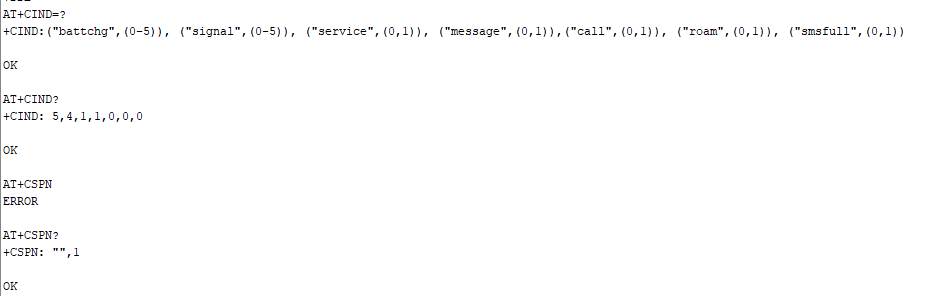
Aşağda bazı komutların serial monitor’de verdiği cevaplar bulunmaktadır

|  |
| --- |
|  |

Şekil-5

|  |
| --- |
| Şekil-6    Şekil-7 |

|  |  |
| --- | --- |
|  |  |



Şekil-8

|  |
| --- |
|  |

Şekil-9

Yukarda gösterilen şekiller’de AT-COMMAND ve ona verilen tipkiler bulunmaktadır

AT-COMMAND komutları ile SIM çipin tipkileri veya ayarlamak istediklerimiz hususları bu komutlar vasıtasıyla ayarlanabilir

Daha fazla bilgiler edinmek için

<https://www.elecrow.com/wiki/images/2/20/SIM800_Series_AT_Command_Manual_V1.09.pdf>

sitesini ziyaret ederek detaylı bilgiler elde edilebilir , ayrıca program kısımında bizim projemizde kullanılacak olan komutlar gösterilecektir

GOOGLE MAPS

Netlify sitesi üzerinde üçretsiz olarak bir site oluşturuldu ve bu siteye githubla birlikte bağlandı böylece web sitesinde istenilen değişikleri gerçekleştirmek için github’ta bulunan kodları değiştirerek site’de bu değişikleri görebilmekteyiz

<https://thirsty-chandrasekhar-b5518a.netlify.com/>

yukardaki linkte tasarlanan site bulunur .

bu platform POST yöntemi kullanarak mikroişlemci’den (sim okuyucusu kullanarak ) GPS bilgilerimi aktarmaya izin verilmedi ,bu yüzden konumu anlık olarak değiştirilmeye fırsatım olmadı .

Keil proramı

Bu bölümde Keil program kullanarak geliştirdiğim kodu ve programın sonuçları paylaşmaktayım .

Program’da yapılan işlemler açıklandı , özet olarak ilk önce kullanılacak olan UART’ları kuruldu daha sonra fonksiyonlar kullanarak gsm bağlantısı yapıldı daha sonra gps ile konum bilgileri alındı .

Aşağdaki Keil programı kullanark yazılan kodlar yazılacaktır ve projeye daha iyi anlaşılacaktır .

////// program'da kullanilacak olan fonksiyonlar

void UART7Tx(char c);

char UART7Rx(void);

char UART5Rx(void);

void UART5Tx(char c);

void delayMs(int n);

unsigned int check=0,i ;

unsigned int check1=0,i1 ;

char info[800];

char info1[120];

char test[6]={"$GPRMC"};

void UART7\_OutString(char \*pt);

void getrx(void);

void UART\_Init(void);

///////////// ANA PROGRAM //////////

int main(void)

{

char c;

//////// UART'lari kurulur //////

SYSCTL->RCGCUART |= 0xFF; /\* provide clock to UART5 and UART0 AND UART1 \*/

SYSCTL->RCGCGPIO |= 0xFF; /\* Enable clock to PORTE AND PORT B AND PORT A\*/

/\* UART5 initialization \*/

UART5->CTL = 0; /\* disable UART5 \*/

UART5->IBRD = 104; /\* 16MHz/16=1MHz, 1MHz/104=9600 baud rate \*/

UART5->FBRD = 11; /\* fraction part, see Example 4-4 \*/

UART5->CC = 0; /\* use system clock \*/

UART5->LCRH = 0x60; /\* 8-bit, no parity, 1-stop bit \*/

UART5->CTL = 0x301; /\* enable UART5, TXE, RXE \*/

/\* UART1 initialization \*/

UART7->CTL = 0; /\* disable UART1 \*/

UART7->IBRD = 104; /\* 50MHz/16=3.125MHz, 3.125MHz/325=9600 baud rate \*/

UART7->FBRD = 11; /\* fraction part, see Example 4-4 \*/

UART7->CC = 0; /\* use system clock \*/

UART7->LCRH = 0x60; /\* 8-bit, no parity, 1-stop bit, no FIFO \*/

UART7->CTL = 0x301; /\* enable UART1, TXE, RXE \*/

/\* UART5 TX5 and RX5 use PE5 and PE4. Set them up. \*/

GPIOE->DEN = 0x33; /\* make PE5, PE4 as digital \*/

GPIOE->AMSEL = 0; /\* turn off analog function \*/

GPIOE->AFSEL = 0x33; /\* use PE5, PE4 alternate function \*/

GPIOE->PCTL = 0x00110011; /\* configure PE5, PE4 for UART5 \*/

UART\_Init();

/// gsm fonksiyonu git ve baglantiyi orda kur ///

UART\_OutChar('\n');

gsm();

UART\_OutChar('\n');

/// gps'i kur ve konum bilgileri al ///

get\_kordinat();}

/// sim800l baglantisini kur ve internet'e baglan ///

int gsm(void)

{

char c;

// c = UART7Rx();

delayMs(1500);

int k=0;

UART7\_OutString("AT\r\n"); //// çipin çalistigindan emin olmak için ///

for(k =0 ; k<8 ;k++)

{

delayMs(1);

c =UART7Rx(); }

UART\_OutChar('\n');

delayMs(100);

UART7\_OutString("AT+CBAND?\r\n"); /// bandin içinde olmasi gerekir ///

for(k =0 ; k<45 ;k++)

{

delayMs(1);

c =UART7Rx(); }

UART\_OutChar('\n');

delayMs(100);

UART7\_OutString("AT+CGATT=1\r\n");

for(k =0 ; k<14;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(100);

UART7\_OutString("AT+CSMINS?\r\n"); /// SIM kartin çipte olmasindan emin olmamiz lazim

for(k =0 ; k<28 ;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

UART7\_OutString("AT+CSPN?\r\n");

for(k =0 ; k<20 ;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

UART7\_OutString("AT+CPAS\r\n");

for(k =0 ; k<18 ;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

UART7\_OutString("AT+CFUN?\r\n"); ///// simin fonkisyonluk durumundan emin olmak için

for(k =0 ; k<26 ;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+COPS?\r\n");

for(k =0 ; k<35 ;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+COPS?\r\n");

for(k =0 ; k<35 ;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+CBC\r\n"); //// enerji durumundan emin olmak için

for(k =0 ; k<27 ;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

delayMs(1000);

UART\_OutChar('\n');

UART7\_OutString("AT+CREG=?\r\n"); /// register durumunu bilmek için

for(k =0 ; k<30 ;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

delayMs(1000);

UART\_OutChar('\n');

UART7\_OutString("AT+CSTT?\r\n");

for(k =0 ; k<30 ;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

UART7\_OutString("AT+CREG?\r\n");

for(k =0 ; k<25 ;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+CREG=1\r\n"); //// sebekeye kayid olmak komutu

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+CREG?\r\n");

for(k =0 ; k<25 ;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+HTTPPARA?\r\n");

for(k =0 ; k<164;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+COPS=1,2,""28601""\r\n"); ///// sebeke'deki bir hosta baglanmak için

delayMs(1000);

UART7\_OutString("AT+COPS?\r\n");

for(k =0 ; k<35;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

///// asagdaki komutlar HTTP BAGLANTISI KURMAK IÇIN GEREKLI OLAN KOMUTLARDIR

UART7\_OutString("AT+SAPBR=3,1,""Contype"",""GPRS""\r\n"); /// http baglantisi hazirlamak için

for(k =0 ; k<23;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+SAPBR=3,1,""APN"",""CMNET""\r\n"); // http baglantisi hazirlamak için

for(k =0 ; k<14;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+SAPBR =1,1\r\n"); // http baglantisi hazirlamak için

for(k =0 ; k<14;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+SAPBR=2,1\r\n"); // http baglantisi hazirlamak için

for(k =0 ; k<14;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+HTTPPARA=""REDIR"",1\r\n");

for(k =0 ; k<14;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+HTTPSSL=0\r\n");

for(k =0 ; k<14;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+HTTPSSL=1\r\n");

for(k =0 ; k<14;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+HTTPPARA=""PROIP"",""148.233.151.240""\r\n");

for(k =0 ; k<15;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+HTTPPARA=""PROPORT"",""8080""\r\n");

for(k =0 ; k<15;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART\_OutChar('\n');

UART\_OutChar('h');

UART\_OutChar('\n');

///// BAGLANTI KURULACAK OLAN WEB SITE ADRESI //////

UART7\_OutString("AT+HTTPPARA=""URL"",""http://api.thingspeak.com/apps/thinghttp/send\_request?api\_key=SHJKOGBNQ14RN5BH""\r\n");

for(k =0 ; k<25;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+HTTPPARA=""CID"",1\r\n");

for(k =0 ; k<23;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

delayMs(1000);

UART7\_OutString("AT+CGATT=1\r\n");

for(k =0 ; k<14;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\"\r\n");

for(k =0 ; k<14;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+SAPBR=3,1,\"APN\",\"\"\r\n");

for(k =0 ; k<14;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+SAPBR=1,1\r\n");

for(k =0 ; k<14;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+HTTPINIT\r\n");

for(k =0 ; k<14;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+HTTPPARA=""URL"",""http://ptsv2.com/t/kbuproject/post""\r\n");

for(k =0 ; k<25;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+HTTPACTION=0\r\n");

for(k =0 ; k<47;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

delayMs(300);

UART7\_OutString("AT+HTTPREAD=0,20\r\n");

for(k =0 ; k<54;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

UART7\_OutString("AT+HTTPPARA?\r\n");

for(k =0 ; k<187;k++)

{

delayMs(1);

c =UART7Rx(); /\* get a character from UART \*/}

UART\_OutChar('\n');

delayMs(1000);

///// KONUMU BILGILERI ELDE EDILMEK IÇINDIR /////

////

int get\_kordinat(void)

{

char c;

for(check =0 ; check<550 ;check++)

{

delayMs(1);

c = UART5Rx(); /\* get a character from UART \*/

info[check++] = c;

}

for(check =0 ; check<600;check++)

{

for(check =0 ; check<600 ;check++)

{

if(info[check]!=test[check])

{

for(check =1 ; check<200 ;check++)

{

UART\_OutChar(info[check]);

}

UART\_OutChar('\n');

delayMs(5000);

// get\_kordinat();

}}}

Yukardaki kodlar projenin önemli kısımlarıdır

Gerekli olan açıklamalar yapıldı

Sonuç ve gerçekleştirilmeyen hususlar

Raporum’da belirtirdiğim gibi projem 100%gerçekleştirilmedi ,özetle yapılan işlemler konum belirleme , mikroişlemcinin internet’e bağlanması ve web sayfasının tasarımı gerçekleştirildi fakat web sayfasına erişim sağlanmadığı için proje tam hedefine ulaşmamıştır.

Proje gerçekleştirirken yaptığım hatalar :

* Daha önce TM4C123GH6PM mikroişlemciyi kullanmadığım için onu öğrenmek çok zamanım aldı bu hatam sayısında karışılacağım yeni bir mikroişlemcinin izlenecek yol öğrenmemdir .
* Başta IME’si kayıtlı olmayan bir sim okuyucusu aldım ve bağlantının kurulmamsına neden olan bu hatayı anlayabilmek için bir çok dene yapmak zorunda kaldım bu’da zaman kayıbını neden oldu.
* Projeyi kısa zaman’da yetişebileceğim sanıyordum ve biraz geç kaldım

Proje’de öğrendiklerim :

* Araştırma yöntemlerim geliştirdi
* TM4C123GH6PM mikroişlemcinin kullanılması
* PCB devre tasarımı
* NMEA sentence
* AT-COMMAND

Referanslar

1. <http://www.ti.com/lit/ds/symlink/tm4c123gh6pm.pdf>
2. <http://www.ti.com/>
3. <https://www.youtube.com/channel/UCVf1i0hXlLsXIrF7V1NWA0Q>
4. <http://users.ece.utexas.edu/~valvano/arm/lectures2.html>
5. <https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf>
6. <http://forum.researchdesignlab.com/datasheet/modules/GSM%20sim900.pdf>
7. <http://www2.keil.com/mdk5/>
8. <https://www.4pcb.com/free-pcb-layout-software/>
9. <http://ijraet.com/pdf10/12.pdf>
10. <https://www.netlify.com/>
11. <https://www.gpsinformation.org/dale/nmea.htm>