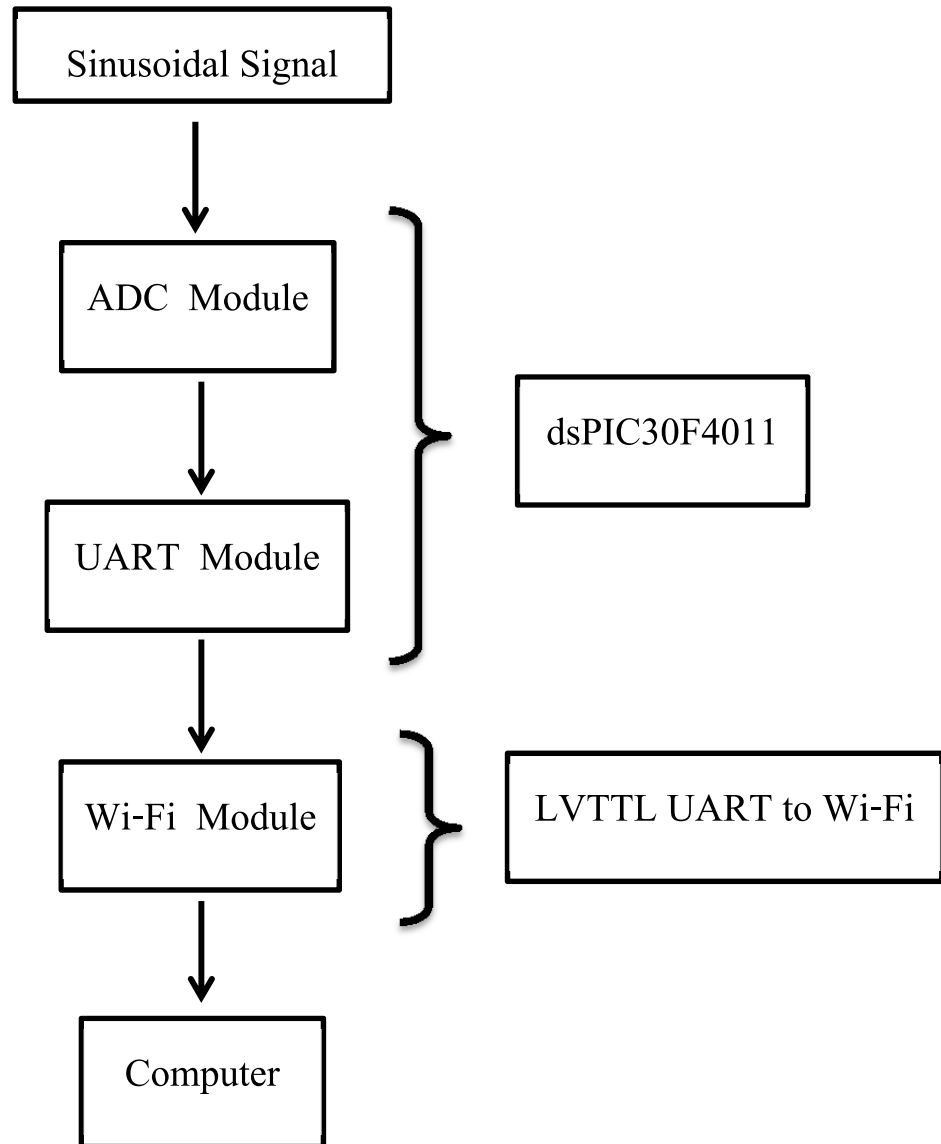


บทที่ 3

การออกแบบและการดำเนินการสร้าง



รูปที่ 3.1 บล็อกการทำงานในการสุ่มสัญญาณและส่งผ่านไวไฟของโครงการ

3.1 การออกแบบเครื่องส่งสัญญาณแสดงผลบนจอมอนิเตอร์ผ่าน WIFI

ในโครงการการส่งสัญญาณอนาล็อกเพื่อนำค่าแรงดันไฟฟ้ามาแสดงผลบนจอมอนิเตอร์โดยผ่าน WIFI ทางผู้จัดทำได้ทำการออกแบบฮาร์ดแวร์และซอฟต์แวร์ โดยทางด้านฮาร์ดแวร์ประกอบด้วยไอซี dsPIC304011 ซึ่งโมดูลที่ใช้ในโครงการนี้ประกอบด้วย โมดูลแปลงสัญญาณอนาล็อกเป็นดิจิทัลและโมดูลสื่อสารข้อมูล UART ฮาร์ดแวร์อีกตัวคือ LVTTTL UART to WIFI เอาไว้รับค่าจากโมดูล UART และส่งไปยัง Computer ตามบล็อกไดอะแกรมรูปที่ 3.1 ทางด้านซอฟต์แวร์จะประกอบด้วย โปรแกรม UART-WIFI เพื่อติดตั้ง Configuration ให้กับ LVTTTL UART to WIFI และโปรแกรม MATLAB รับค่าที่ส่งมาจากฮาร์ดแวร์และทำการแสดงผลออกมาทางจอมอนิเตอร์

3.2 โมดูลภายในไอซี dsPIC30F4011

3.2.1 โมดูลแปลงสัญญาณอนาล็อกเป็นดิจิทัลและโมดูล UART กับพอร์ตอนุกรม

ทำหน้าที่รับสัญญาณอนาล็อกซึ่งจะถูกป้อนเข้ามาที่อินพุตขา AN0 ผ่านช่องส่งสัญญาณ CH0 จากนั้นจะเข้าขบวนการแปลงโดยการใช้ Sample/Hold โดยโปรแกรมจะเก็บ (Sample) และคง (Hold) ค่าแรงดันเอาไว้ตลอดช่วงทำงานจนกว่าจะแปลงเสร็จ เมื่อแปลงเสร็จแล้วค่าที่ได้จะถูกนำไปเก็บไว้ในตัวแปร ADCBUF0 - ADCBUF0 ของไอซี จากนั้นเลือกอ่านค่าที่เก็บไว้ใน Buffer ที่เราตั้งไว้ คือ ADCBUF0 รับข้อมูลจาก ADCBUF0 เพื่อส่งต่อไปยังอุปกรณ์ภายนอกด้วยความเร็ว 115200 bps ด้วยขา Tx ของ dsPIC30F4011

3.3 โมดูลไวไฟ LVTTTL UART to Wi-Fi (IEEE 802.11b/g)

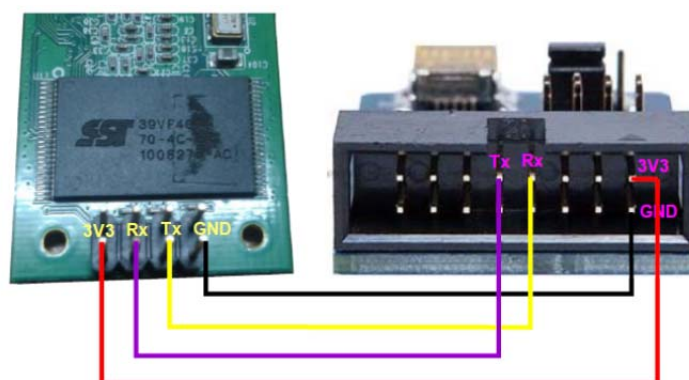


รูปที่ 3.2 ด้านหน้าและด้านหลังของ LVTTTL UART to Wi-Fi

LVTTL UART to Wi-Fi คือ โมดูลที่ใช้สำหรับแปลงการรับส่งข้อมูลในรูปแบบ UART เป็นการรับส่งข้อมูลในรูปแบบของ Wireless LAN หรือ Wi-Fi (IEEE 802.11b/g) ซึ่งภายในโมดูลมี Software TCP/IP Stack อยู่ทำให้ใช้งานได้ง่าย สะดวก และรวดเร็ว เหมาะสำหรันำมาใช้กับระบบประมวลผลหรือไมโครคอนโทรลเลอร์ที่มีความเร็วในการประมวลผลต่ำ ใช้หน่วยความจำน้อย และลดเวลาในการพัฒนาลง โดยโมดูลมีการจัดสรร Software ที่เป็นส่วนของ TCP/IP Stack และการควบคุม Hardware ที่ใช้รับส่งข้อมูลผ่านทาง Wireless ไว้แล้ว ดังนั้นเพียงแค่ตั้งค่าตัวโมดูลให้สามารถเชื่อมต่อกับระบบ Network ผ่านทาง Software ที่ผู้ผลิตได้จัดเตรียมไว้ให้ก็สามารถใช้สื่อสารข้อมูลได้เลย นอกจากนี้ยังสามารถสั่งงานหรือเปลี่ยนแปลงค่าต่างๆ ผ่านทาง AT Command ด้วยการเขียนโปรแกรมจากไมโครคอนโทรลเลอร์ได้อีกด้วย

3.4 การตั้งค่าโมดูล LVTTL UART to Wi-Fi (IEEE 802.11b/g)

3.4.1 การเชื่อมต่อโมดูล LVTTL UART to Wi-Fi กับ โมดูล USB to Serial



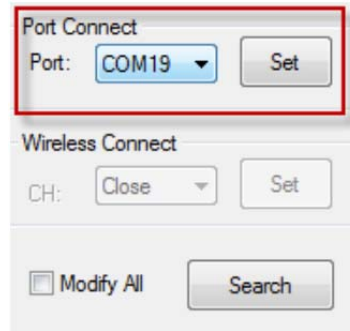
รูปที่ 3.3 การเชื่อมต่อ LVTTL UART to Wi-Fi เข้ากับ USB To Serial



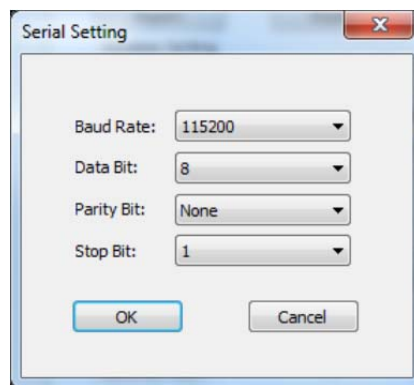
รูปที่ 3.4 Software Uart – WIFI

การตั้งค่าเบื้องต้นทางผู้จัดทำได้ทำการตั้งค่าโมดูล LVTTL UART to Wi-Fi พร้อมกับ โมดูล USB To Serial เพื่อความสะดวกในการติดตั้ง การเชื่อมต่อนั้นทำได้โดยเชื่อมขา Tx ของโมดูล USB To Serial เข้ากับขา Rx ของโมดูล LVTTL UART to Wi-Fi และขา Rx ของโมดูล USB To Serial เข้ากับขา Tx ของโมดูล LVTTL UART to Wi-Fi จากนั้นเชื่อมไฟเลี้ยงและกราวด์เข้าด้วยกันดังรูปที่ 3.3 พร้อมกับการเปิด Software Uart – Wi-Fi เพื่อตั้งค่าต่าง ๆ ให้กับโมดูล ดังรูปที่ 3.4

3.4.2 การตั้งค่า Port Connect ภายใน Software Uart – WIFI



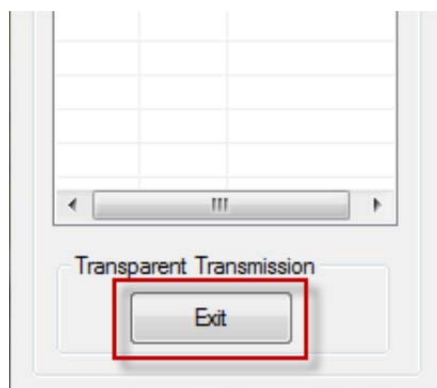
รูปที่ 3.5 เลือก Comport ใน Software UART



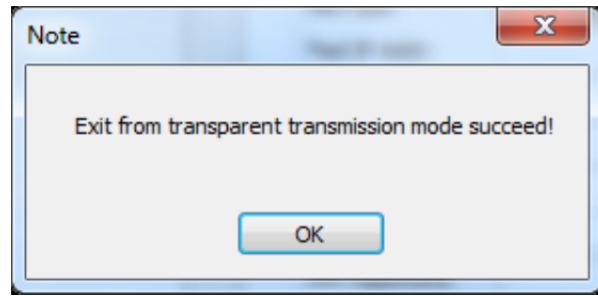
รูปที่ 3.6 หน้าต่าง Serial Setting

เมื่อเปิดโปรแกรม Uart – Wi-Fi อันดับแรกเลือก Comport ดังรูปที่ 3.5 ที่ทำการเชื่อมต่อไว้กับโมดูล จากนั้นคลิกที่ Button Set จะปรากฏหน้าต่าง Serial Setting ภายในจะปรากฏค่า Baud Rate , Data Bit , Parity Bit และ Stop Bit ให้เราเลือกค่าได้ตามที่เราต้องการ แต่ผมได้ทำการตั้งค่าไว้คือ Baud Rate = 115200 , Data Bit = 8 , Parity = None และ Stop Bit = 1 และกดปุ่ม OK ดังรูปที่ 3.6

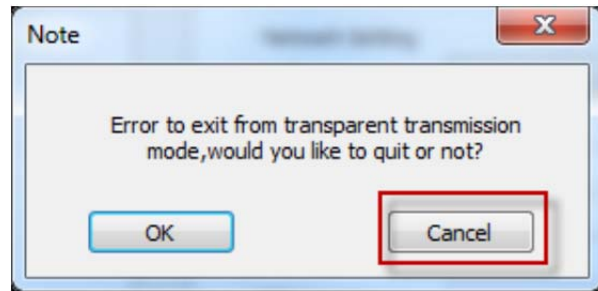
3.4.3 การออกจาก Auto Work Mode เพื่อติดต่อกับโมดูล



รูปที่ 3.7 การกดปุ่ม Exit เพื่อออกจากการทำงานแบบ Auto Work Mode



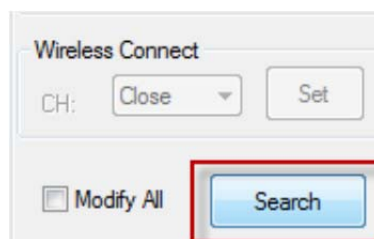
รูปที่ 3.8 การออกจาก Auto Work mode สำเร็จ



รูปที่ 3.9 การออกจาก Auto Work mode ไม่สำเร็จ

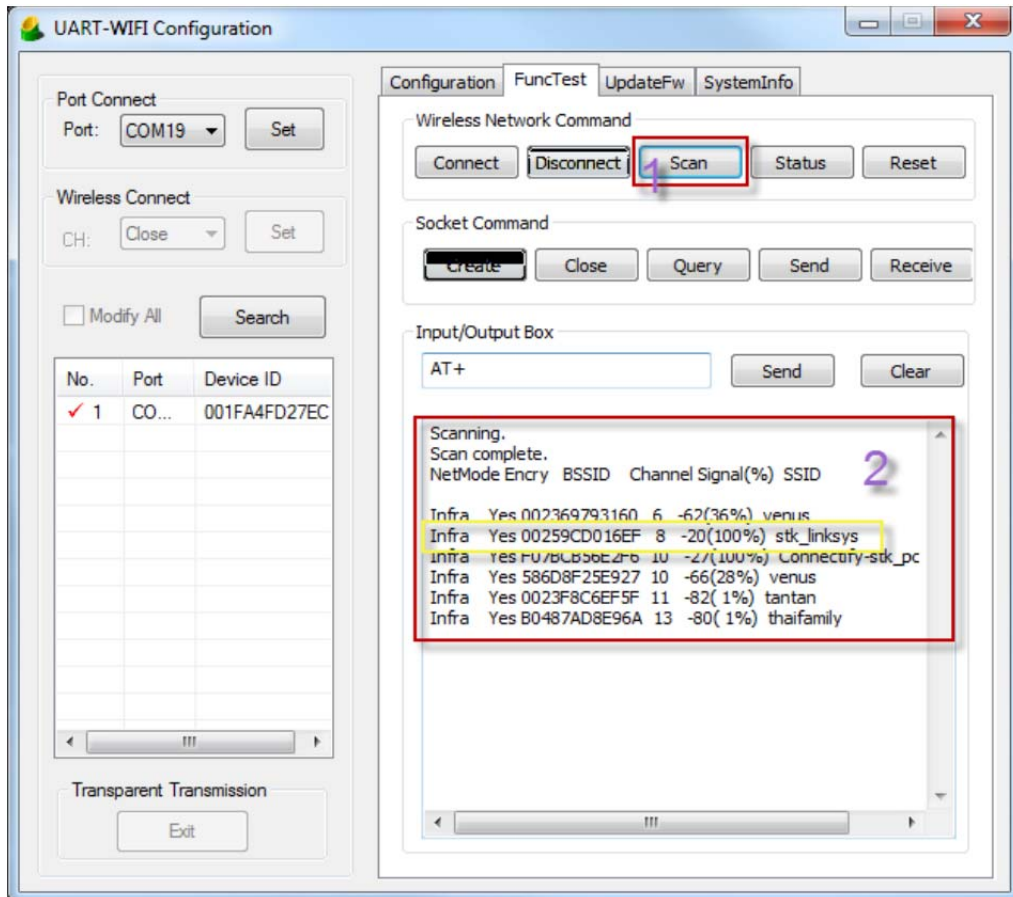
เมื่อตั้งค่าจากข้อ 3.4.2 เสร็จแล้ว ให้ทำการกดปุ่ม Exit เพื่อต้องการจะออกจาก Auto Work Mode เพื่อติดต่อกับโมดูล ดังรูปที่ 3.7 หากออกจาก Auto Work mode สำเร็จจะปรากฏหน้าต่างดังรูปที่ 3.8 และถ้าหากออกจาก Auto Work mode ไม่สำเร็จจะปรากฏหน้าต่างดังรูปที่ 3.9

3.4.4 การค้นหา Port ที่ทำการเชื่อมต่อกับโมดูล



รูปที่ 3.10 กดปุ่ม Search เพื่อค้นหา Port ที่ทำการเชื่อมต่ออยู่

3.4.5 การค้นหา Access Point ในระยะของโมดูล



รูปที่ 3.12 เลือก Tab FuncTest จากนั้น scan หา Access point

เมื่อเลือก Port ที่ทำการเชื่อมต่อได้แล้วนั้น ไปที่ Tab Configuration ทำการ Scan เพื่อค้นหา Access Point ที่จะทำการเชื่อมต่อด้วย ดังรูปที่ 3.12 การทดลองนี้จะทำการเชื่อมต่อโมดูลเข้ากับ Access Point ที่ชื่อ stk_linksys โดย AP stk_linksys มีรายละเอียดดังนี้

SSID = stk_linksys

Encrypt Modes = WPA2(AES)

Key = “stk123456”

IP Address = 192.168.3.1

3.4.6 การตั้งค่า Wireless Setting และ Network Setting ให้กับโมดูล

Wireless Setting

SSID:

Encry Mode: Key Type:

Key Index: 1 2 3 4

Key:

รูปที่ 3.13 การตั้งค่า Wireless Setting ให้กับตัวโมดูล

Network Setting

NetType:

Fixed IP Addr:

Subnet Mask:

Gateway Addr:

DNS Addr:

PPP User Name:

PPP PassWord:

รูปที่ 3.14 การตั้งค่า Network Setting ให้กับตัวโมดูล

จากรายละเอียดของ Access Point นำมากรอกในช่อง Wireless Setting ดังรูปที่ 3.13 และ Network Setting ทำการเลือก NetType เป็น Static ip จากนั้นกรอกค่าตามรูปที่ 3.14

3.4.7 การตั้งค่า Work Mode Setting ให้กับโมดูล

Workmode Setting

Auto Workmode

ProtocolType: C/S Mode:

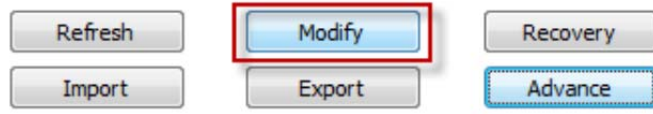
ServerAddr: Port:

TCPConTime: s(0~10000000,0 means forever)

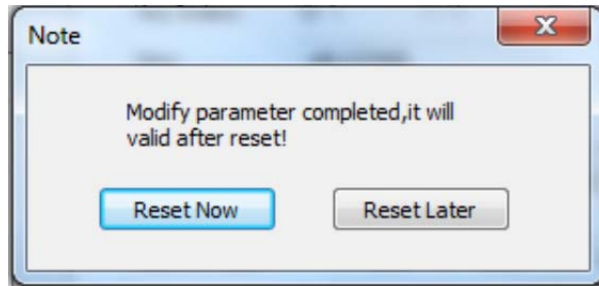
รูปที่ 3.15 การตั้งค่าใน Work mode Setting

การตั้งค่าการใช้งานในโหมดนี้ คือ เมื่อโมดูลเริ่มทำงานตัวโมดูลจะ Connect ไปยัง Access Point ที่เรากำหนดค่าเอาไว้คือ AP stk_linksys โดยตัวโมดูลจะทำงานเป็น TCP Client และเมื่อมี Data เข้ามาทาง Serial ของโมดูล ข้อมูลนั้นจะถูกส่งผ่าน Wireless LAN ไปยัง Server ที่กำหนดเอาไว้คือ IP 192.168.3.102 ที่ Port 8000

3.4.8 การปรับแต่งค่าของโมดูลตามที่ได้ตั้งค่าไว้



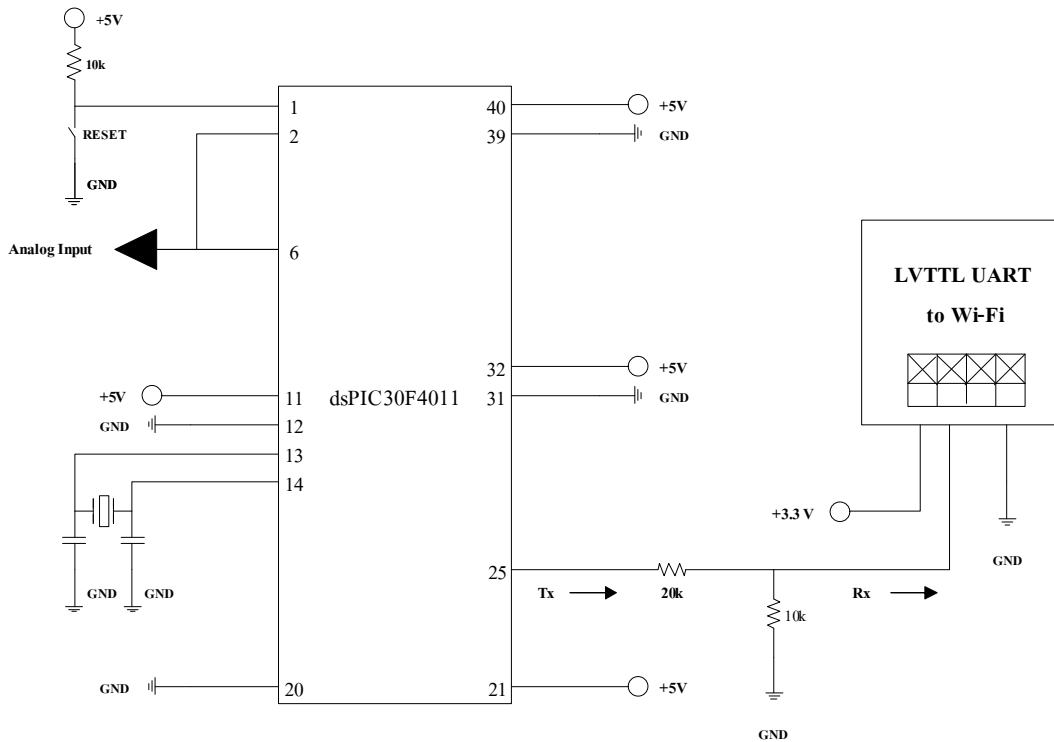
รูปที่ 3.16 การปรับค่า Setting โดยการเลือก Button Modify



รูปที่ 3.17 เลือก Restart Now หลังจาก Modify

เมื่อตั้งค่าตามขั้นตอนข้างบนเรียบร้อยแล้ว ก็มาถึงขั้นตอนสุดท้ายคือ ปรับแต่งค่าทั้งหมดให้กับโมดูล กดที่ปุ่ม Modify ดังรูปที่ 3.16 จากนั้น Message Box จะปรากฏขึ้นมา ให้เราเลือก Restart Now ตามรูปที่ 3.17 ก็จบขั้นตอนการตั้งค่าให้กับโมดูล LVTTTL UART to Wi-Fi

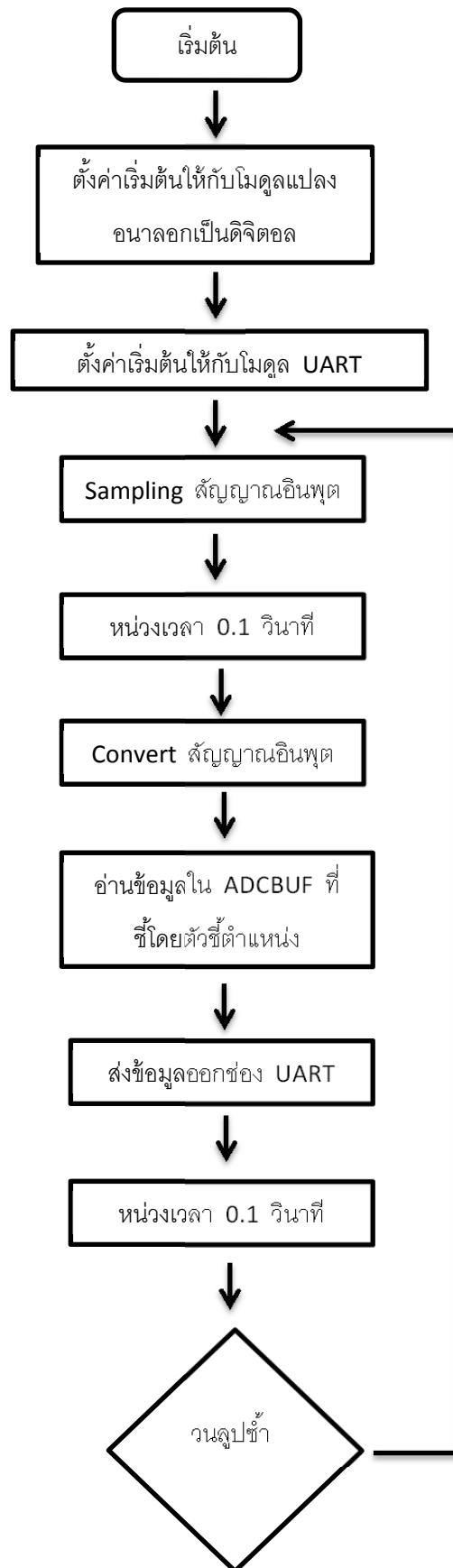
3.5 การออกแบบ dsPIC30F4011 ส่ง Data ผ่าน TCP ไปยัง Client



รูปที่ 3.18 แสดง Schematic ของวงจร

จากที่อธิบายไว้เบื้องต้นเรื่องโมดูลแปลงสัญญาณอนาลอกเป็นดิจิทัลและโมดูลการสื่อสารข้อมูลแบบ UART จะแสดง Schematic ดังรูปที่ 3.18 ขาที่ 1 ต่อไฟ +5 V เข้ากับรีจิสเตอร์ 10 โอห์ม ผ่านสวิทช์กดติด-ปล่อยดับ เพื่อทำปุ่ม RESET จากนั้นต่อไฟ +5 V เข้าที่ขา 11 , 21 , 32 และ 40 และกราวด์ของระบบเข้าที่ขา 12 , 20 , 31 และ 39 ขาที่ 13 และ 14 ต่อคริสตัลความถี่ 7.3728 MHz และใน dsPIC30F4011 สามารถใช้ PLL ความถี่ RUN ได้ถึง 29.4912 MHz ขาที่ 25 เป็นขาส่งข้อมูลที่ใช้วิธีการสื่อสารแบบ UART ผลลัพธ์ที่ออกมาจะมีค่าตามแรงดันที่อ้างอิง 0-5 V แต่ตัวโมดูล LVTTTL UART to Wi-Fi นั้นรับแรงดันได้เพียง 3.3 V เพราะฉะนั้นขาที่ 25 จึงจำเป็นต้องผ่าน Voltage Divider เพื่อลดแรงดันของข้อมูลที่ออกจากขาที่ 25 เหลือ 3.3 V เพื่อเข้าขาที่ 2 ของ LVTTTL UART to Wi-Fi

Flow Chart แสดงการทำงานภายใน dsPIC30F4011



จาก Flow Chart สามารถนำมาเขียนเป็น Code ได้ดังนี้

```
#include <p30f4011.h>
#include<adc10.h>
#include<uart.h>
#include<stdio.h
#include<stdlib.h>
_FOSC(CSW_FSCM_OFF & XT_PLL4);
_FWDT(WDT_OFF);
#define Fcy                7372800.0
#define BAUD_RATE          115200.0
#define BAUD_RATE_GEN      (Fcy/(16*BAUD_RATE))-1;
void _ISR_U1TXInterrupt(void)
{
    _U1TXIF = 0;
}
void _ISR_U1RXInterrupt(void)
{
    _U1RXIF = 0;
}

void delay_ms(unsigned int ms){
    unsigned int I;
    for(; ms>0; ms--)
        for(i=0; i<728; i++)
            Nop();
}

void uart1_init()
{
    unsigned int ConfigU1MODE,ConfigU1STA,BaudRate;
    CloseUART1();
    ConfigU1MODE =    UART_EN &
                    UART_IDLE_CON &
                    UART_RX_TX&
```

```

        UART_DIS_WAKE &
        UART_DIS_LOOPBACK &
        UART_DIS_ABAUD &
        UART_NO_PAR_8BIT &
        UART_1STOPBIT;
ConfigU1STA = UART_INT_TX_BUF_EMPTY &
        UART_TX_PIN_NORMAL &
        UART_INT_RX_CHAR&
        UART_ADR_DETECT_DIS &
        UART_RX_OVERRUN_CLEAR;

    BaudRate = BAUD_RATE_GEN;
    OpenUART1(ConfigU1MODE, ConfigU1STA, BaudRate);
}

void adc_init()
{
    unsigned int Channel, PinConfig, Scanselect, Adcon3_reg, Adcon2_reg, Adcon1_reg;
    ADCON1bits.ADON = 0;
    Channel = ADC_CH0_POS_SAMPLEA_AN0 &
        ADC_CH0_NEG_SAMPLEA_NVREF ;
    SetChanADC10(Channel);
    ConfigIntADC10(ADC_INT_DISABLE);
    PinConfig = ENABLE_AN0_ANA ;
    Scanselec = SKIP_SCAN_AN4 ;

    ADCON3bits.SAMC = 0;
    ADCON3bits.ADCS = 4;
    Adcon3_reg = ADC_CONV_CLK_INTERNAL_RC ;
    Adcon2_reg = ADC_VREF_AVDD_AVSS &
        ADC_SCAN_ON &
        ADC_ALT_BUF_OFF &
        ADC_ALT_INPUT_OFF &
        ADC_CONVERT_CH0&

```

```

        ADC_SAMPLES_PER_INT_1;

    Adcon1_reg = ADC_MODULE_ON &
        ADC_IDLE_CONTINUE &
        ADC_FORMAT_INTG &
        ADC_CLK_MANUAL &
        ADC_SAMPLE_SIMULTANEOUS &
        ADC_AUTO_SAMPLING_ON;

    OpenADC10(Adcon1_reg, Adcon2_reg, Adcon3_reg, PinConfig, Scanselct);
}

int main(void)
{
    unsigned int value1 , value2;
    adc_init();
    uart1_init();
    {
        ADCON1bits.SAMP = 1;
        delay_ms(100);
        ConvertADC10();
        while(BusyADC10());

        value1 = &ADCBUF0;
        delay_ms(100);
        putcUART1(value1);
        while (BusyUART1());
        delay_ms(100);
        value2 = value1>>=8;
        putcUART1(value2);
        while (BusyUART1());
    }
    return 0;
}

```

3.6 การออกแบบการรับและแสดงผลข้อมูลบนฝั่ง Client

เมื่อรู้ไอพีของตัว Server และส่งมาในรูปแบบ TCP แล้วก็สามารออกแบบการรับข้อมูลบนโปรแกรม MATLAB ของฝั่ง Client เพื่อรับข้อมูลและนำมาแสดงผลตาม Code ที่ทำการออกแบบไว้ดังนี้

```
t = tcpip('192.168.11.10', 4567, 'NetworkRole', 'client')
fopen(t)
data=fread(t, [1,100],'uint8')
fclose(t)
```