Abenezer Namaga
CPE 403

Github root directory: https://github.com/Ayertena/TIVAC

TIVAC the Light

GOAL:

- Configure I2c interface between TM4C123GXL board and TSL2951 sensor
- Configure UART communication between TM4C123GXL and ESP8266 module
- Upload TSL2951 data readings to cloud server hosted on thingspeak.com

DELIVERABLES:

The final project was supposed to record luminosity from the TSL2951 sensor and send that data via I2C to the TM4C123G; that light data was then processed and transmitted to the ESP8266 Wi-Fi module which would connect to the cloud and graph the data. All of this was accomplished with the most difficult part being configuring the TSL2951 sensor. I was not properly initializing the ALS portion of the sensor, and not leaving a delay for the values to be gathered so I would only get 0. After fixing this I was able to get values as low as 0 (pitch black) or as high as 1138.83.

COMPONENTS:

- **Tiva TM4C123GH6PM –** MCU to control project. Device is initialized by setting the system clock, enabling GPIO module, enabling the I2C module, enabling and setting the clock for I2C0 master module, enabling UART1, and configuring port pins.
- **TSL2591 –** A high dynamic digital light sensor which can measure the Lux up to ranges of 188u – 88,000 Lux. Its interfaced with the TivaC using I2C.
- **ESP8266 -** A Wi-Fi module interfaced with the TivaC using UART. It is initialized and will send information to server.
- **TSL2591 –** A high dynamic digital light sensor which can measure the Lux up to ranges of 188u – 88,000 Lux. Its interfaced with the TivaC using I2C. It is initialized by reading the Device ID register setting the gain, the timing, and the Power On Register to 1. It will read the channel data from Channel0 and Channel1; moreover, these two values are calculated to find the Lux value of the sensor and the value is returned to the caller.

SCREENSHOTS & PHOTO:

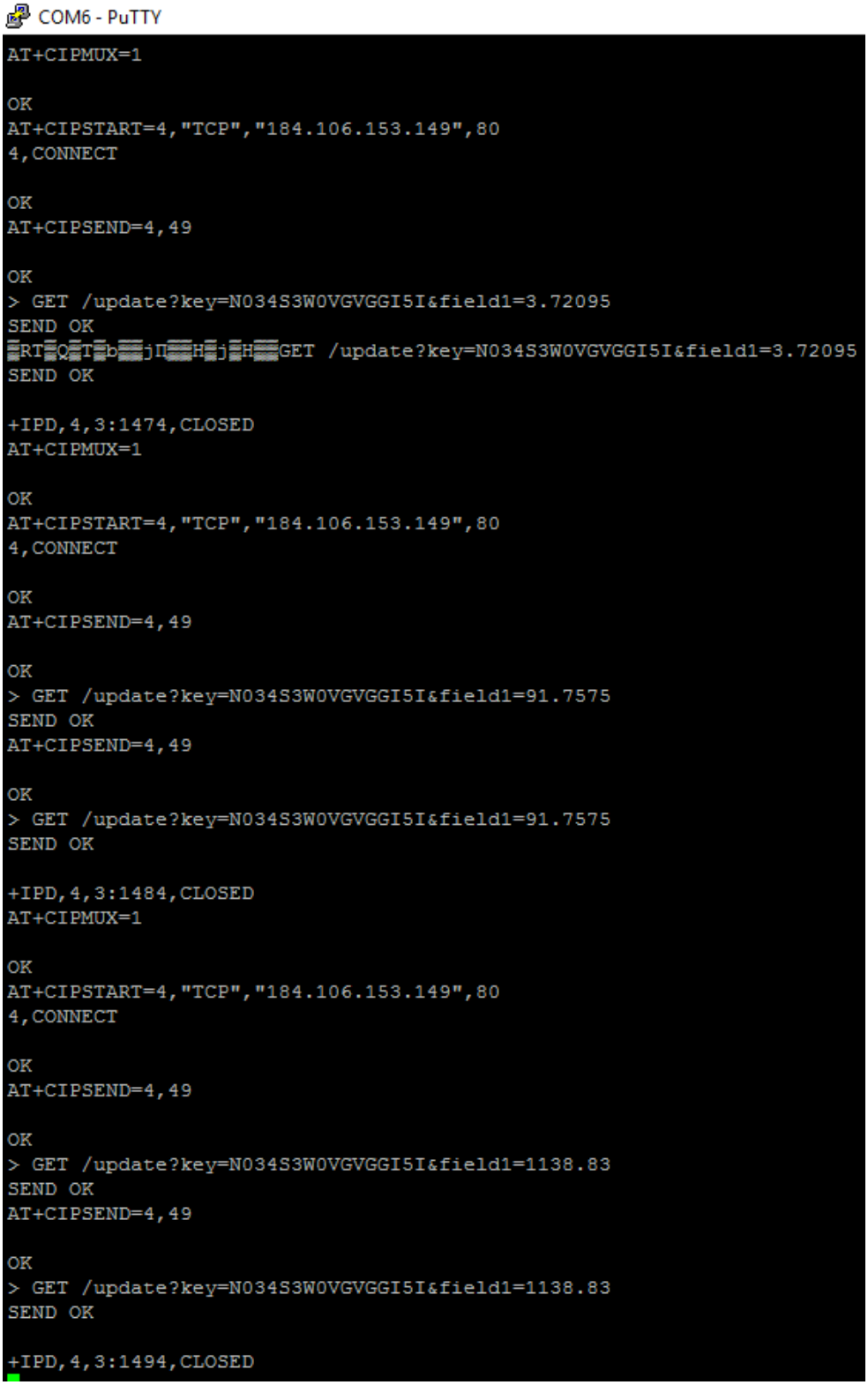


Figure 1- Putty terminal showing strings being transmitted from TM4C to ESP8266 shows 3 posts being made with low, medium, and high lighting on senso

Updated: less than a minute ago
Last entry: less than a minute ago
Entries: 161

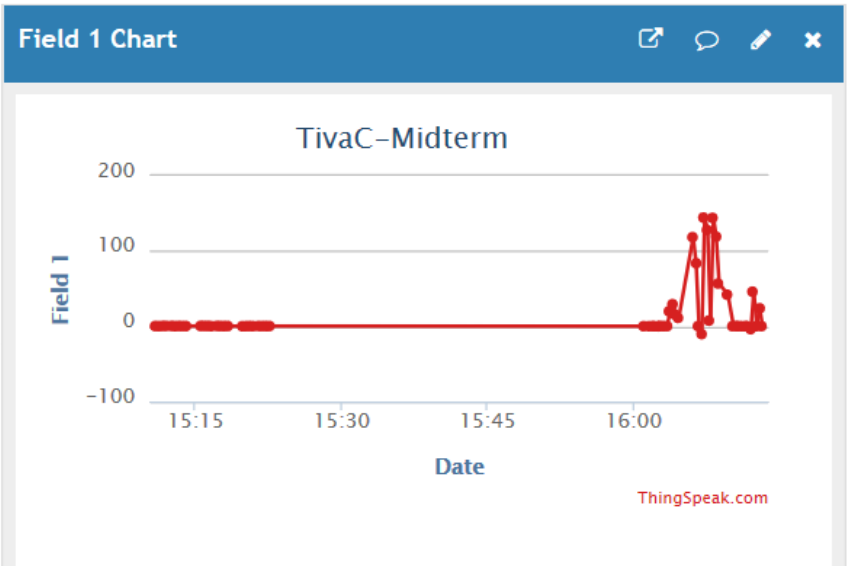


Figure 2- Thingspeak plot of data from TSL2591 (Peaks represent bright light contact with sensor)

ESP8266

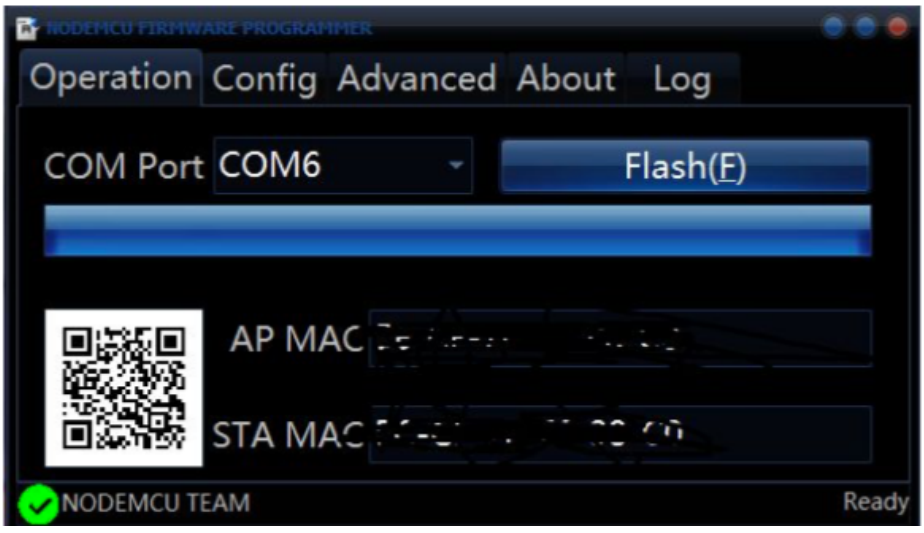


Figure 2- ESP8266 flash success

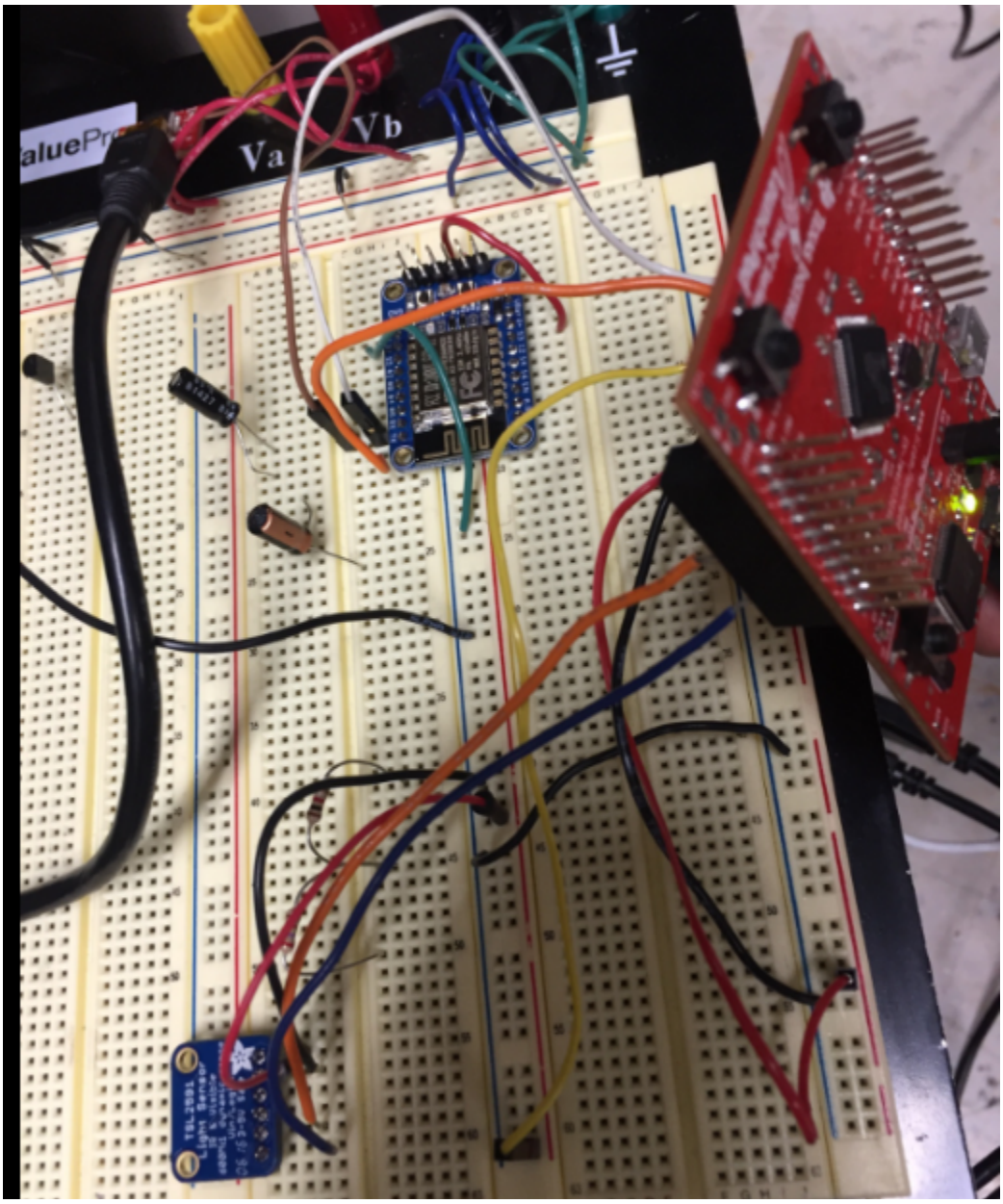
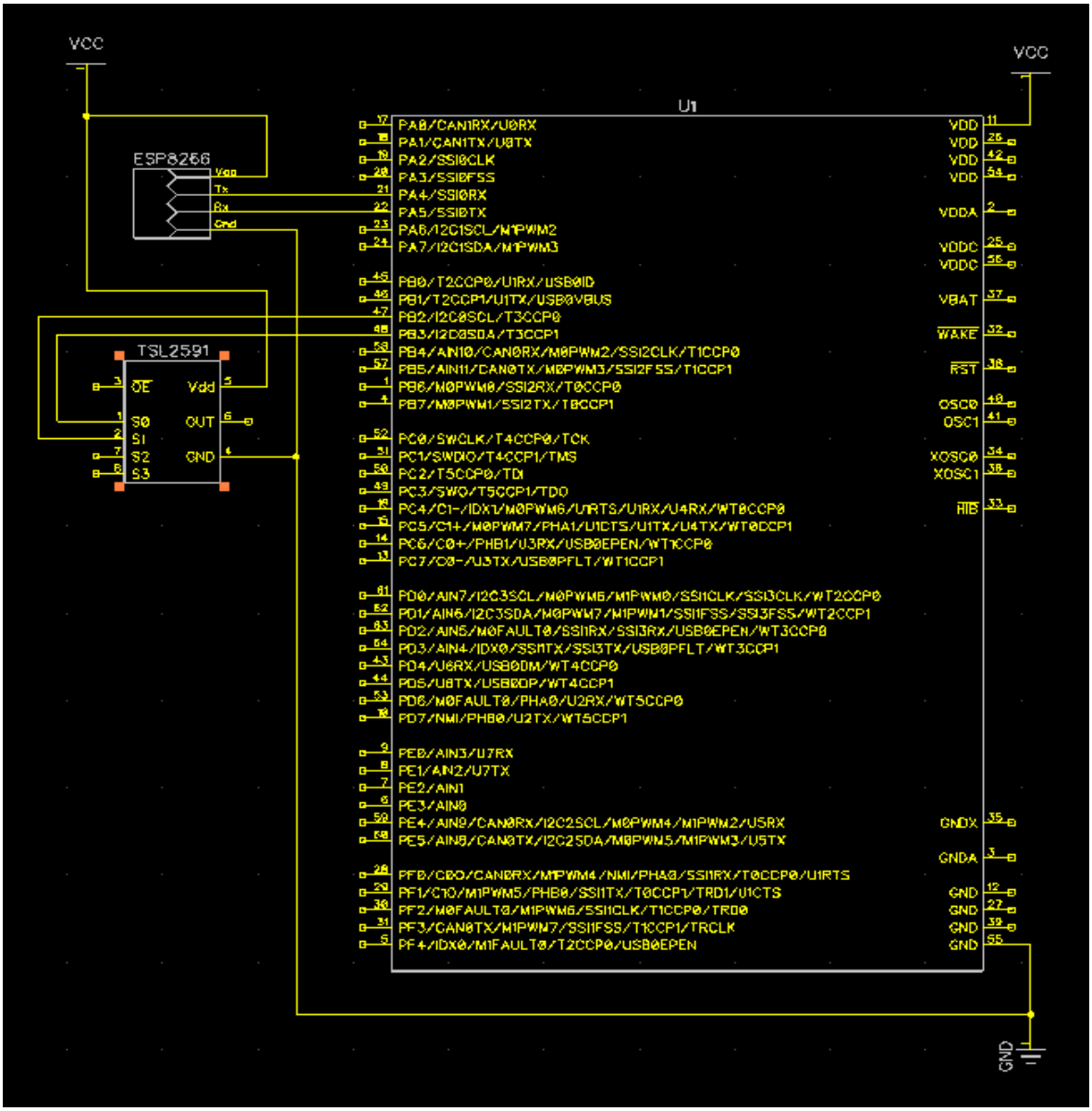


Figure 4- Photo showing interconnection of TM4C, TSL2951, and ESP8266

SCHEMATICS:


VCC
VCC
U1
ESP8266
Vcc
Tx
Rx
Gnd
TSL2591
OE
Vdd
S0
OUT
S1
S2
GND
S3
PA0/CAN1RX/U0RX
PA1/CAN1TX/U0TX
PA2/SSI0CLK
PA3/SSI0FSS
PA4/SSI0RX
PA5/SSI0TX
PA6/I2C1SCL/M1PWM2
PA7/I2C1SDA/M1PWM3
PB0/T2CCP0/U1RX/USB0ID
PB1/T2CCP1/U1TX/USB0VBUS
PB2/I2C0SCL/T3CCP0
PB3/I2C0SDA/T3CCP1
PB4/AIN10/CAN0RX/M0PWM2/SSI2CLK/T1CCP0
PB5/AIN11/CAN0TX/M0PWM3/SSI2FSS/T1CCP1
PB6/M0PWM0/SSI2RX/T0CCP0
PB7/M0PWM1/SSI2TX/T0CCP1
PC0/SWCLK/T4CCP0/TCK
PC1/SWDIO/T4CCP1/TMS
PC2/T5CCP0/TDI
PC3/SWO/T5CCP1/TDO
PC4/C1-/IDX1/M0PWM6/U1RTS/U1RX/U4RX/WT0CCP0
PC5/C1+/M0PWM7/PHA1/U1CTS/U1TX/U4TX/WT0CCP1
PC6/C0+/PHB1/U3RX/USB0EPEN/WT1CCP0
PC7/C0-/U3TX/USB0PFLT/WT1CCP1
PD0/AIN7/I2C3SCL/M0PWM6/M1PWM0/SSI1CLK/SSI3CLK/WT2CCP0
PD1/AIN6/I2C3SDA/M0PWM7/M1PWM1/SSI1FSS/SSI3FSS/WT2CCP1
PD2/AIN5/M0FAULT0/SSI1RX/SSI3RX/USB0EPEN/WT3CCP0
PD3/AIN4/IDX0/SSI1TX/SSI3TX/USB0PFLT/WT3CCP1
PD4/U6RX/USB0DM/WT4CCP0
PD5/U6TX/USB0DP/WT4CCP1
PD6/M0FAULT0/PHA0/U2RX/WT5CCP0
PD7/NMI/PHB0/U2TX/WT5CCP1
PE0/AIN3/U7RX
PE1/AIN2/U7TX
PE2/AIN1
PE3/AIN0
PE4/AIN9/CAN0RX/I2C2SCL/M0PWM4/M1PWM2/U5RX
PE5/AIN8/CAN0TX/I2C2SDA/M0PWM5/M1PWM3/U5TX
PF0/C0O/CAN0RX/M1PWM4/NMI/PHA0/SSI1RX/T0CCP0/U1RTS
PF1/C1O/M1PWM5/PHB0/SSI1TX/T0CCP1/TRD1/U1CTS
PF2/M0FAULT0/M1PWM6/SSI1CLK/T1CCP0/TRD0
PF3/CAN0TX/M1PWM7/SSI1FSS/T1CCP1/TRCLK
PF4/IDX0/M1FAULT0/T2CCP0/USB0EPEN
VDD
VDD
VDD
VDD
VDDA
VDDC
VDDC
VBAT
WAKE
RST
OSC0
OSC1
XOSC0
XOSC1
RBB
GNDX
GNDA
GND
GND
GND
GND
GND

CODE:

```
#include <stdbool.h>
#include <stdint.h>
#include "inc/hw_i2c.h"
#include "inc/hw_ints.h"
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/gpio.h"
#include "driverlib/i2c.h"
#include "driverlib/interrupt.h"
#include "driverlib/pin_map.h"
#include "driverlib/sysctl.h"
#include "driverlib/uart.h"
#include "utils/uartstdio.h"
#include "driverlib/rom_map.h"
#include "inc/hw_types.h"
#include "inc/hw_gpio.h"

uint32_t ui32SysClock;


const uint8_t TSL2591address = 0x29;


//UART1
void InitConsole(void)
{
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);
    GPIOPinConfigure(GPIO_PB0_U1RX);
    GPIOPinConfigure(GPIO_PB1_U1TX);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_UART1);
    UARTClockSourceSet(UART1_BASE, UART_CLOCK_PIOSC);
    GPIOPinTypeUART(GPIO_PORTB_BASE, GPIO_PIN_0 | GPIO_PIN_1);
    UARTStdioConfig(1, 115200, 16000000);
}

void ftoa(float f,char *buf)
{
       /*Function acquired from forum:
        *
http://e2e.ti.com/support/microcontrollers/stellaris_arm/f/471/p/44193/156824.aspx
        */

      /*Parses through float # and stores the value as a character array*/
    int pos=0,ix,dp,num;
    if (f<0)
    {
        buf[pos++]='-';
        f = -f;
    }
    dp=0;
    while (f>=10.0)
    {
```

```
        f=f/10.0;
        dp++;
    }
    for (ix=1;ix<8;ix++)
    {
            num = (int)f;
            f=f-num;
            if (num>9)
                buf[pos++]='#';
            else
                buf[pos++]='0'+num;
            if (dp==0) buf[pos++]='.';
            f=f*10.0;
            dp--;
    }
}


// USING I2C0 - CHECK FOR TIVAC LAUNCHPAD PINS
void i2c0_init()
{
             //enable i2c0
       MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_I2C0);
       //reset i2c0
       MAP_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);
       //Set pin type for PB3, configure as data pin
       MAP_GPIOPinTypeI2C(GPIO_PORTB_BASE, GPIO_PIN_3);
       MAP_GPIOPinConfigure(GPIO_PB3_I2C0SDA);

       //Set pin type for PB2, configure as SCL
       MAP_GPIOPinTypeI2CSCL(GPIO_PORTB_BASE, GPIO_PIN_2);
       MAP_GPIOPinConfigure(GPIO_PB2_I2C0SCL);

       //Enable and initialize i2c0 master module. Use System Clock.  Set i2C data
rate at 100kbps (try true for 400)
       I2CMasterInitExpClk(I2C0_BASE, SysCtlClockGet(), false);
              while (I2CMasterBusy(I2C0_BASE));

}

void i2c0_write(uint8_t dev_addr, uint8_t dev_reg, uint16_t dev_data)
{

      //Tell device we want to write to bus
    I2CMasterSlaveAddrSet(I2C0_BASE, dev_addr, false);
    //Bitwise or command bit to tell device we want to write to dev_reg
    I2CMasterDataPut(I2C0_BASE, TSL2591_COMMAND_BIT |dev_reg);
      //Send and wait
    I2CMasterControl(I2C0_BASE, I2C_MASTER_CMD_BURST_SEND_START);
       while (I2CMasterBusy(I2C0_BASE));
       //Now write the data to the register
    I2CMasterSlaveAddrSet(I2C0_BASE, dev_addr, true);
      //place the data in Master and send
    I2CMasterDataPut(I2C0_BASE, dev_data);
    I2CMasterControl(I2C0_BASE, I2C_MASTER_CMD_BURST_RECEIVE_FINISH);
```

```
        while (I2CMasterBusy(I2C0_BASE));

}


uint8_t i2c0_read(uint32_t slave_addr, uint8_t reg)
{
    //specify that we are writing (a register address) to the
    //slave device
    I2CMasterSlaveAddrSet(I2C0_BASE, slave_addr, false);
    while(I2CMasterBusy(I2C0_BASE));
    //specify register to be read
    I2CMasterDataPut(I2C0_BASE, TSL2591_COMMAND_BIT | reg);
    while(I2CMasterBusy(I2C0_BASE));
    //send control byte and register address byte to slave device
    I2CMasterControl(I2C0_BASE, I2C_MASTER_CMD_SINGLE_SEND);

    //wait for MCU to finish transaction
    while(I2CMasterBusy(I2C0_BASE));
    //specify that we are going to read from slave device
    I2CMasterSlaveAddrSet(I2C0_BASE, slave_addr, true);
    while(I2CMasterBusy(I2C0_BASE));
    //send control byte and read from the register we
    //specified
    I2CMasterControl(I2C0_BASE, I2C_MASTER_CMD_SINGLE_RECEIVE);
    //wait for MCU to finish transaction
    while(I2CMasterBusy(I2C0_BASE));
    //return data pulled from the specified register
    return I2CMasterDataGet(I2C0_BASE);
}

void TSL2591_init()
{
        uint8_t DevID;
        i2c0_write(TSL2591address,TSL2591address,1); //literally no idea what this
does


        DevID = i2c0_read(TSL2591address,TSL2591_REGISTER_DEVICE_ID); //read device ID
        //Alert user if errors reading ID

       // Set Gain and Timing
        i2c0_write(TSL2591address,0x01, 0xB0 );//CONFIG medium gain and reset the
device (self-clearing)
        i2c0_write(TSL2591address, 0x00, 0x03);//ENABLE Power, Oscillator,

}

void TSL2591_disable()
{
      i2c0_write(TSL2591address,0xA0, 0x00); //turn off TSL2591
}

float getLuminosity ()
{
```

```
  float    atime = 100.0F, again=25.0F; //For 100ms integration time and med gain
  float    cpl, lux1, lux2, lux;
  uint16_t ch0 ;
  uint16_t ch1 ;

// Get full luminosity
  //read channel 0 into x0
  uint16_t x0;
  uint16_t y0;


  y0 = i2c0_read(TSL2591address,0x14);
  while(I2CMasterBusy(I2C0_BASE));
 // UARTprintf("\n\nChannel 0L: %d", y0);
  x0 = i2c0_read(TSL2591address,0x15); //C0DataHigh
  while(I2CMasterBusy(I2C0_BASE));
 // UARTprintf("\nChannel 0H: %d", x0);
  x0 <<= 8;


  uint16_t x1;
  uint16_t y1;
  //read channel 1 into x1
  y1 = i2c0_read(TSL2591address, 0x16);
  while(I2CMasterBusy(I2C0_BASE));
 // UARTprintf("\nChannel 1L: %d", y1);
  x1 = i2c0_read(TSL2591address, 0x17);
  while(I2CMasterBusy(I2C0_BASE));
 // UARTprintf("\nChannel 1H: %d", x1);
  x1 <<= 8;

// Disable Sensor


  ch0 = x0+y0;
  ch1 = x1+y1;
//Calculate Lux value from sensor
       cpl = (atime * again) / TSL2591_LUX_DF;
  lux1 = ( (float)ch0 - (TSL2591_LUX_COEFB * (float)ch1) ) / cpl;
  lux2 = ( ( TSL2591_LUX_COEFC * (float)ch0 ) - ( TSL2591_LUX_COEFD * (float)ch1 ) )
/ cpl;
  lux = lux1 > lux2 ? lux1 : lux2;

  return lux;
}

void ESPSENDPREP()
{
       SysCtlDelay(2000000);
       UARTprintf("AT+CIPMUX=1\n\r");
       SysCtlDelay(2000000);
       UARTprintf("AT+CIPSTART=4,\"TCP\",\"184.106.153.149\",80\n\r");
       SysCtlDelay(2000000);
//     SysCtlDelay(20000000);
//     UARTprintf("GET /update?key=N034S3W0VGVGGI5I&field1=50\n\r");
```

```
//     SysCtlDelay(20000000);
//     UARTprintf("GET /update?key=N034S3W0VGVGGI5I&field1=\n\r");

}

int main(void)
{
      char myLux[48] = "GET /update?key=N034S3W0VGVGGI5I&field1=";//52 bit string,
42 instr, 6 are lux. (append \n\r at end)
    float lux_read;
  ui32SysClock = MAP_SysCtlClockFreqSet((SYSCTL_XTAL_25MHZ |
                 SYSCTL_OSC_MAIN | SYSCTL_USE_PLL |
                 SYSCTL_CFG_VCO_480), 120000000);
  InitConsole();
  i2c0_init();

  while(1)
  {

      // UARTprintf("AT\n");

       TSL2591_init();
       SysCtlDelay(2000000);
       lux_read = getLuminosity();
       ftoa(lux_read, &myLux[40]);
       myLux[47]='\0';
TSL2591_disable();
       ESPSENDPREP();

            UARTprintf("AT+CIPSEND=4,49\n\r");
            SysCtlDelay(20000000);
       UARTprintf("%s\n\r",&myLux);
       SysCtlDelay(20000000);
            UARTprintf("AT+CIPSEND=4,49\n\r");
            SysCtlDelay(20000000);
       UARTprintf("%s\n\r",&myLux);



       SysCtlDelay(5000000*20); //slightly more than 15 seconds per so
  }
}
```

Name: Abenezer Namaga