Brian Lopez
Phillip Sortomme

CPE403 project

Brian Lopez
Phillip Sortomme

CPE403 project

Goal

- Main Goal
  - *transmit data from lux sensor and relay the data to a BeagleBone Black running the TI-Stack application*
- Objectives
  - *lux sensor data was collected via I2C to sensor CC1350 uc node*
  - *connection was establish between the sensor and collector CC1350 uc node*
  - *use collector node as coprocessor to BBB*

Outcome - Accomplishments

- *the sensor node was configured to the TSL2591 lux sensor via I2C and Lux data was transmitted to the collector node*
- *established the collector node as a coprocessor to the BBB*

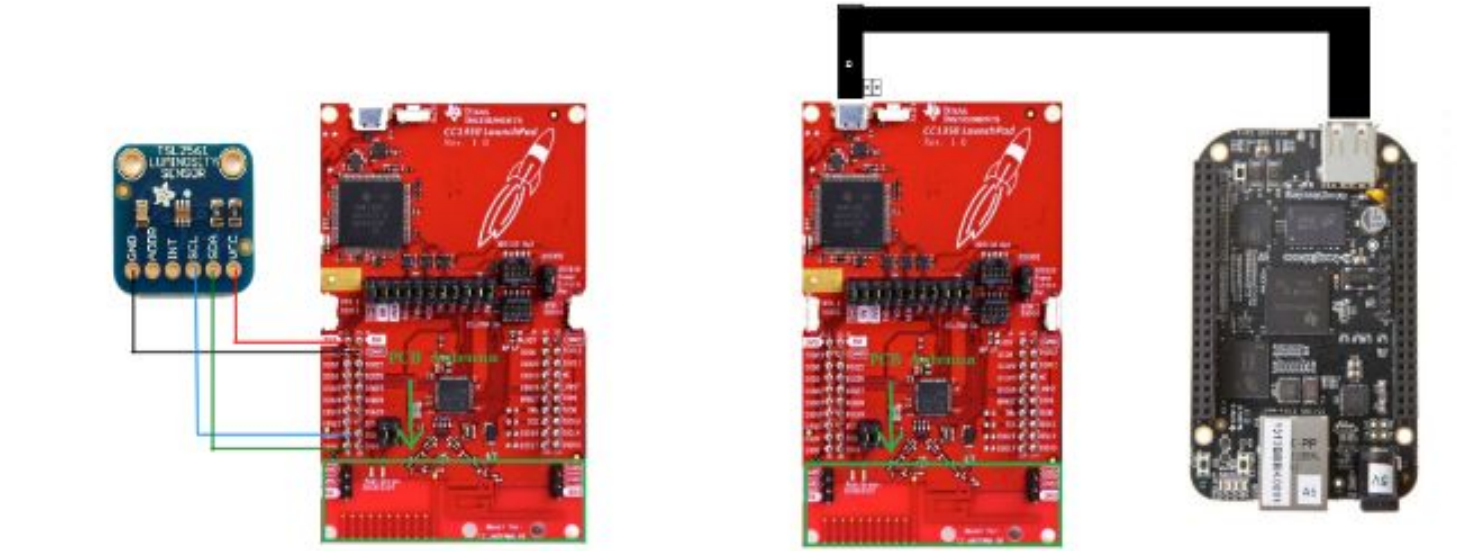
Components Used in Design

- CC1350 uc (2)
- BeagleBone Black
- TSL2591 lux sensor

Tools used in Design

- Code composer Studio
- Uniflash

Schematics



Pre-requisites used in Design

- TI 15.4-Stack Linux Gateway SDK
- TI processor SDK SD card image file
- CCS CC1350 collector/sensor project files

Implementation Details

- Steps used in design:

1 add I2C libraries and enable I2C on sensor node

```
// added to add I2C communication
#include <ti/drivers/I2C.h>
#include <ti/drivers/i2c/i2CCC26XX.h>
#include "board.h"
// Not needed for I2C, only for debugging
// Used to display text/values through UART
#include "board_lcd.h"

uint8_t         txBuffer[5]; // holds the commands being sent
uint8_t         rxBuffer[5]; // holds anything sent from sensor
I2C_Handle      i2c;
I2C_Params      i2cParams;
I2C_Transaction i2cTransaction;

I2C_init();          // Configure I2C on pins 4 and 5
I2C_Params_init(&i2cParams); // set up the parameters
i2c = I2C_open(Board_I2C_TMP, &i2cParams); // establish I2C
```

Implementation Details

- 2 configure Lux sensor

```
i2cTransaction.writeCount = 2;
txBuffer[0] = 0xA1; // Register control | Command bit
txBuffer[1] = 0x10;
i2cTransaction.readCount = 0;
if (I2C_transfer(i2c, &i2cTransaction)) { // I2C_transfer sends the data
    // sends the two registers to lux sensor
}


txBuffer[0] = 0xA0; // Register Enable | Command bit
txBuffer[1] = 0x8B; //enable poweron, aen, aien, npien
if (I2C_transfer(i2c, &i2cTransaction)) { // I2C_transfer sends the data
    // sends next two registers to finish configuration
}
```

```
    uint32_t x1;
    i2cTransaction.writeCount = 2;
    txBuffer[0] = 0xB4; // Command bit | C0DataH
    txBuffer[1] = 0xB5; // Command bit | C0DataL
    i2cTransaction.readCount = 2; // indicate that we are reading 2 values
    if (I2C_transfer(i2c, &i2cTransaction)) {
        x1 = rxBuffer[0];  // get C0DataH
        x1 <<= 16;
        x1 |= rxBuffer[1]; // get C0DataL

        x1 /= 2500;    // no lux calculation, just divide the raw value
    }

    /* Deinitialized I2C */
    I2C_close(i2c);
    // Display the lux value through UART, only for debugging
    LCD_WRITE_STRING_VALUE("Lux is: ", (uint16_t) x1, 10, 5);
    tempSensor.objectTemp = (uint16_t) x1;
```

Implementation Details

- 3 verify lux data transmission by reading transmitted data on collector node
- 4 flash Collector node with default CC1350 SDK build using Uniflash
- 5 flash BBB with Linux Arago then install TI Stack gateway.

Actual project set-up

sensor node

collector node

Demo

testing I2C transmission of lux data
https://youtu.be/NHI1rWbbn-k

testing integration with BBB
https://youtu.be/3QfJmNJaPts

Results and Conclusions

lux values from I2C transmission

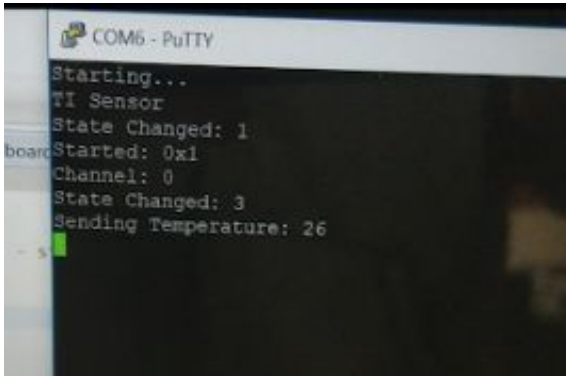
sensor node

collector node

Results and Conclusions

Linux gateway implementation using default temperature code. This was used to test correct connections. In the end we were unable to combine the two portions to transmit lux values to the BBB.

sensor node



BBB Stack application of Collector node

Reference

Datasheets:

TSL2591: https://cdn-shop.adafruit.com/datasheets/TSL25911_Datasheet_EN_v1.pdf
CC1350: http://www.ti.com/lit/ds/swrs183b/swrs183b.pdf
BeagleBone Black: https://cdn-shop.adafruit.com/datasheets/BBB_SRM.pdf

Goal

- Main Goal
  - *transmit data from lux sensor and relay the data to a BeagleBone Black running the TI-Stack application*
- Objectives
  - *lux sensor data was collected via I2C to sensor CC1350 uc node*
  - *connection was establish between the sensor and collector CC1350 uc node*
  - *use collector node as coprocessor to BBB*

Outcome - Accomplishments

- *the sensor node was configured to the TSL2591 lux sensor via I2C and Lux data was transmitted to the collector node*
- *established the collector node as a coprocessor to the BBB*

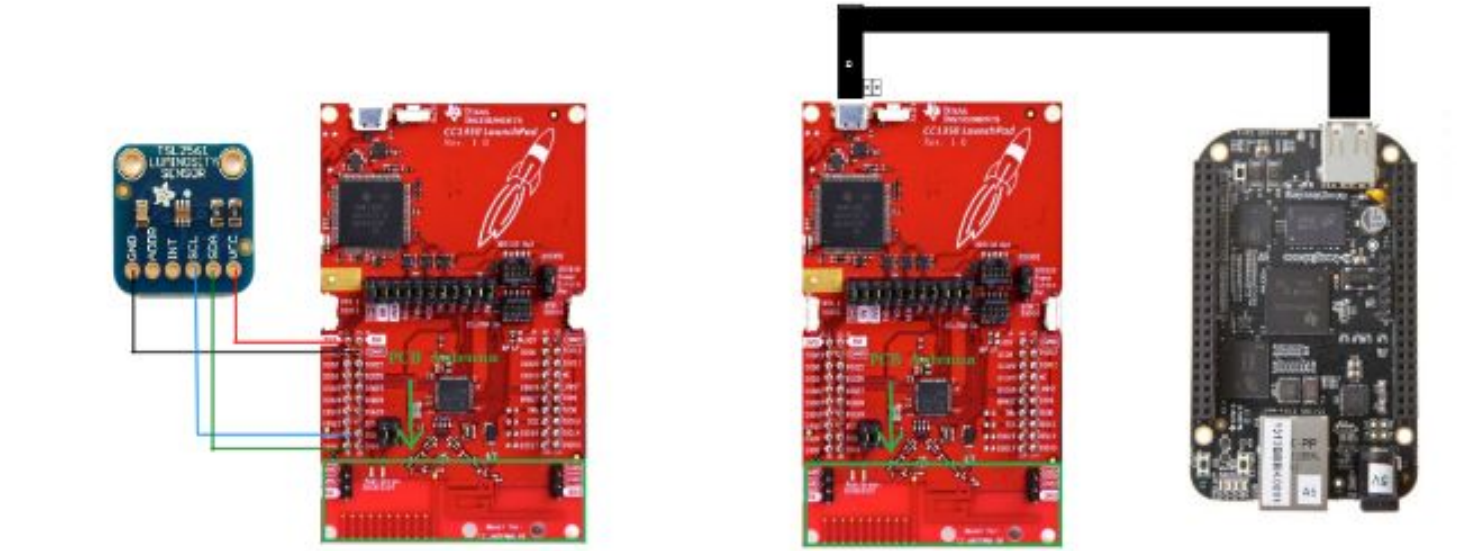
Components Used in Design

- CC1350 uc (2)
- BeagleBone Black
- TSL2591 lux sensor

Tools used in Design

- Code composer Studio
- Uniflash

Schematics



Pre-requisites used in Design

- TI 15.4-Stack Linux Gateway SDK
- TI processor SDK SD card image file
- CCS CC1350 collector/sensor project files

Implementation Details

- Steps used in design:

1 add I2C libraries and enable I2C on sensor node

```
// added to add I2C communication
#include <ti/drivers/I2C.h>
#include <ti/drivers/i2c/i2CCC26XX.h>
#include "board.h"
// Not needed for I2C, only for debugging
// Used to display text/values through UART
#include "board_lcd.h"

uint8_t         txBuffer[5]; // holds the commands being sent
uint8_t         rxBuffer[5]; // holds anything sent from sensor
I2C_Handle      i2c;
I2C_Params      i2cParams;
I2C_Transaction i2cTransaction;

I2C_init();          // Configure I2C on pins 4 and 5
I2C_Params_init(&i2cParams); // set up the parameters
i2c = I2C_open(Board_I2C_TMP, &i2cParams); // establish I2C
```

Implementation Details

- 2 configure Lux sensor

```
i2cTransaction.writeCount = 2;
txBuffer[0] = 0xA1; // Register control | Command bit
txBuffer[1] = 0x10;
i2cTransaction.readCount = 0;
if (I2C_transfer(i2c, &i2cTransaction)) { // I2C_transfer sends the data
    // sends the two registers to lux sensor
}


txBuffer[0] = 0xA0; // Register Enable | Command bit
txBuffer[1] = 0x8B; //enable poweron, aen, aien, npien
if (I2C_transfer(i2c, &i2cTransaction)) { // I2C_transfer sends the data
    // sends next two registers to finish configuration
}
```

```
    uint32_t x1;
    i2cTransaction.writeCount = 2;
    txBuffer[0] = 0xB4; // Command bit | C0DataH
    txBuffer[1] = 0xB5; // Command bit | C0DataL
    i2cTransaction.readCount = 2; // indicate that we are reading 2 values
    if (I2C_transfer(i2c, &i2cTransaction)) {
        x1 = rxBuffer[0];  // get C0DataH
        x1 <<= 16;
        x1 |= rxBuffer[1]; // get C0DataL

        x1 /= 2500;    // no lux calculation, just divide the raw value
    }

    /* Deinitialized I2C */
    I2C_close(i2c);
    // Display the lux value through UART, only for debugging
    LCD_WRITE_STRING_VALUE("Lux is: ", (uint16_t) x1, 10, 5);
    tempSensor.objectTemp = (uint16_t) x1;
```

Implementation Details

- 3 verify lux data transmission by reading transmitted data on collector node
- 4 flash Collector node with default CC1350 SDK build using Uniflash
- 5 flash BBB with Linux Arago then install TI Stack gateway.

Actual project set-up

sensor node

collector node

Demo

testing I2C transmission of lux data
https://youtu.be/NHI1rWbbn-k

testing integration with BBB
https://youtu.be/3QfJmNJaPts

Results and Conclusions

lux values from I2C transmission

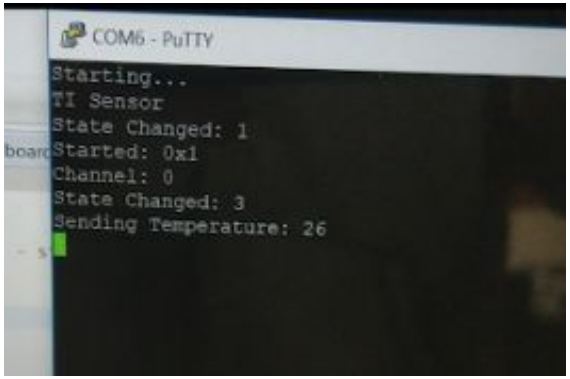
sensor node

collector node

Results and Conclusions

Linux gateway implementation using default temperature code. This was used to test correct connections. In the end we were unable to combine the two portions to transmit lux values to the BBB.

sensor node



BBB Stack application of Collector node

Reference

Datasheets:

TSL2591: https://cdn-shop.adafruit.com/datasheets/TSL25911_Datasheet_EN_v1.pdf
CC1350: http://www.ti.com/lit/ds/swrs183b/swrs183b.pdf
BeagleBone Black: https://cdn-shop.adafruit.com/datasheets/BBB_SRM.pdf