Programming Wiced with Wiced SDK

e-Yantra Team

July 23, 2016

Contents

1 Prerequisites

- You need to have an account on www.cypress.com inorder to download wiced sdk.
- A windows / linux or mac running machine.
- Knowledge about C programming.

2 Hardware Requirement

- Wiced Sense Tag
- A local system (Laptop or PC with bluetooth 4.0 or above)

3 Software Requirement

- Wiced Sense SDK

4 What is Wiced Smart SDK?

- Wiced Smart Software Development Kit is typically a set of software development tools that allows the creation of applications for Wiced Sense device.
- It is Eclipse based IDE enables one-click build and download for Broadcom 2073x BLE device applications.
- Wiced Smart IDE

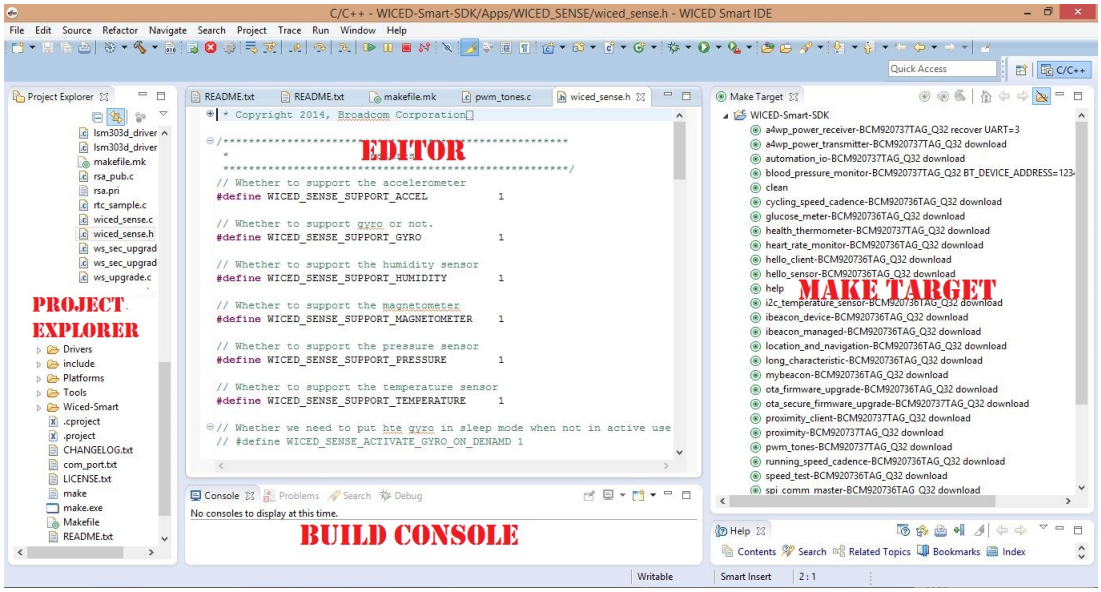


Figure 1: Wiced Sense SDK

- Project Explorer - This section provides list of all the projects that have been created.It contains all the c , h and makefile requird to build the project.
- Editor - This window provides space to write the code according to our project. The code is written in C language.
- Make Target - It creates a target for your device. When you click on the specified target, that code begins to compile and is flashed to your device.
- Build Console - It shows the current status while compiling and flashing the code to the device.

5 Creating a new project

- Download the project from the link given here

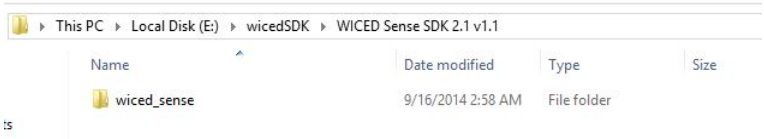


Figure 2: Downloading the wiced code

- Copy the project to the apps directory of wiced sense.
- Open > New > Folder.

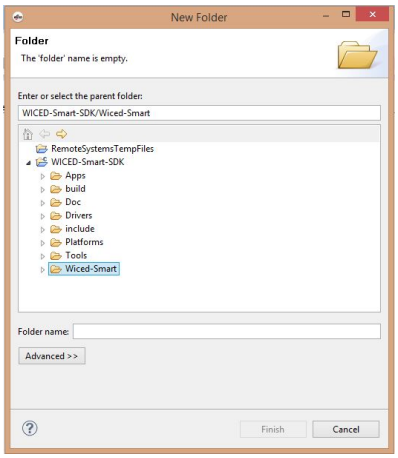


Figure 3: Creating new project

- Insert the name of parent folder to the path where you have placed the wiced sense in apps.

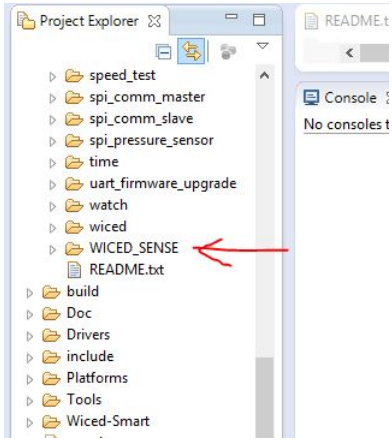
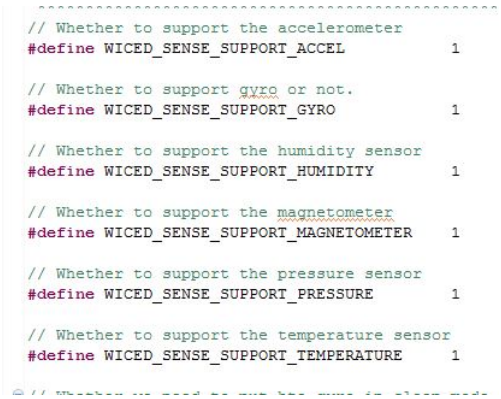


Figure 4: Project Explorer

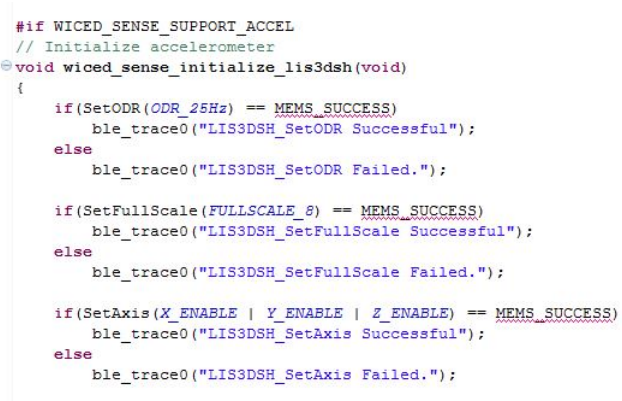
- When you click OK you will see a project named Wiced Sense created in project explorer section



```
// Whether to support the accelerometer
#define WICED_SENSE_SUPPORT_ACCEL           1

// Whether to support gyro or not.
#define WICED_SENSE_SUPPORT_GYRO            1

// Whether to support the humidity sensor
#define WICED_SENSE_SUPPORT_HUMIDITY        1

// Whether to support the magnetometer
#define WICED_SENSE_SUPPORT_MAGNETOMETER    1

// Whether to support the pressure sensor
#define WICED_SENSE_SUPPORT_PRESSURE        1

// Whether to support the temperature sensor
#define WICED_SENSE_SUPPORT_TEMPERATURE     1
```

Figure 5: Setting sensor ON/OFF

- Now you can Set the sensor to be ON/OFF by making the bit in the above code ON or OFF.



```
#if WICED_SENSE_SUPPORT_ACCEL
// Initialize accelerometer
void wiced_sense_initialize_lis3dsh(void)
{
    if(SetODR(ODR_25Hz) == MEMS_SUCCESS)
        ble_trace0("LIS3DSH_SetODR Successful");
    else
        ble_trace0("LIS3DSH_SetODR Failed.");

    if(SetFullScale(FULLSCALE_8) == MEMS_SUCCESS)
        ble_trace0("LIS3DSH_SetFullScale Successful");
    else
        ble_trace0("LIS3DSH_SetFullScale Failed.");

    if(SetAxis(X_ENABLE | Y_ENABLE | Z_ENABLE) == MEMS_SUCCESS)
        ble_trace0("LIS3DSH_SetAxis Successful");
    else
        ble_trace0("LIS3DSH_SetAxis Failed.");
```

Figure 6: Setting different parameters of sensor

- We can set the refresh rate to take the sensor data at particular interval. Now it is set to 25Hz as ODR register is set to 25Hz.
- We can set the full scale range of a particular sensor. Right Now it is set to 8g. It can be changed to 2g,4g,8g,16g.
- We can also make the particular axis of the sensors ON/OFF.
- Now the value from this function is sent to the driver code of that sensor.

- Control register to set the fullscale.

CTRL_REG5 (24h)

Control register 5.

Table 25. Control register 5

BW2	BW1	FSCALE2	FSCALE1	FSCALE0	ST2	ST1	SIM

Table 26. Control register 5 description

BW2:BW1	Anti-aliasing filter bandwidth. Default value: 00 00=800 Hz; 01=400 Hz; 10:=200 Hz; 11:=50 Hz)
FSCALE2:0	Full-scale selection. Default value: 00 000=+/- 2G; 001=+/- 4G; 010=+/- 6G; 011=+/- 8G; 100=+/- 16G
ST2:1	Self-test enable. Default value: 00 00=self-test disabled;
SIM	SPI serial interface mode selection. Default value: 0 0=4-wire interface; 1:=3-wire interface

Figure 7: Control Register for accelerometer

- 100 is passed through value variable which sets the fullscale range to 8g.

```
/*******************************************************************************
* Function Name  : SetFullScale
* Description    : Set FullScale by typedef definition
* Input          : FULLSCALE_2/FULLSCALE_4/FULLSCALE_8/FULLSCALE_16
* Output         : None
* Return         : Status [MEMS_ERROR, MEMS_SUCCESS]
*******************************************************************************/
status_t SetFullScale(Fullscale_t fs) {
  u8_t value;

  if( !ReadReg(MEMS_I2C_ADDRESS, CNTL5, &value) )
    return MEMS_ERROR;

  value &= 0xC7;
  value |= (fs<<FSCALE);

  if( !WriteReg(MEMS_I2C_ADDRESS, CNTL5, value) )
    return MEMS_ERROR;

  return MEMS_SUCCESS;
}
```

Figure 8: Setting Accelerometer full scale

- Now we will make a new target file which will enable us to flash the code.

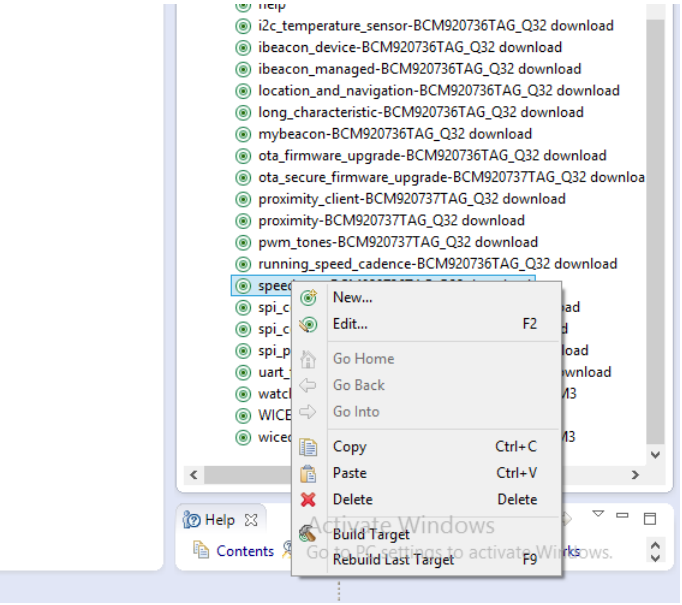


Figure 9: Creating new target

- Give the name of the target file same as that in apps section i.e WICED-SENSE-BCM920737TAG-Q32 download for me.

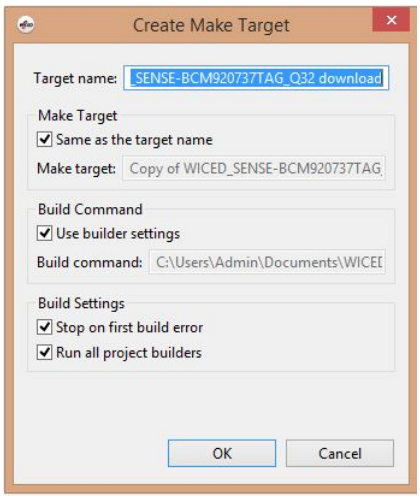


Figure 10: Renaming the target

- Double click the created target file and it will start compiling the code.
- Current process can be observed in console.
- You will get Application Running at the end, it means that you have flashed the code successfully.

6 Reference

https://community.cypress.com/docs/DOC-1759
https://community.cypress.com/docs/DOC-1766