GSM Click - Cross-Platform Development Quick Start Guide

Overview

This document explains how to use the MikroElektronika GSM Click board as an accessory to the following host development platforms (microcontrollers):

- Atmel's SAMA5D3 Xplained
- Atmel's SAMA5D4 Xplained
- Beaglebone Black
- RIoTboard

The GSM Click board contains a Telit GL865-QUAD GSM/GPRS module allowing users to connect their hardware to a GSM cellular network. Sample C programs for each platform are presented.

GSM Click Hardware

The board connects to the host platform via the mikroBUS connector. Communications with the GL865 are by a simple asynchronous serial connection. By default, the serial port connection configuration between the microcontroller and GL865 is 115,200 bps, 8 bits by word, no parity, 1 stop bit. Data are "logic level" voltages (0V, 3.3V) and *not* RS-232 levels (-12V, +12V).

Hardware flow control is affectively disabled by holding the clear-to-send (CTS) line low (enabled). In the example program, the request-to-send (RTS) signal is ignored by the GL865.

The module hardware is reset with a digital input. At the GL865 chip itself, the reset line is active low (pull to ground to reset the chip), but the GSM Click module inverts this signal. The GSM Click module hardware is reset by pulsing the reset line from 0 (ground) to 1 (logic high, 3.3V) and back to 0.

Details for the GL865-QUAD module are found in the Data Sheets and Application Notes (http://www.telit.com/products/product-service-selector/product-service-selector/show/product/gl865-quad/).

SIM Cards

To fully utilize the module, a subscriber identification module (SIM) card must be installed and activated. The SIM card ties the module to the cellular provider's network, billing information, etc., and is necessary for making calls, sending text messages, etc.

For basic testing of the GSM Click Board, the microcontroller can communicate with the module, query the module status, etc. with no SIM card installed.

Software

AT Commands

Control and configuration commands are exchanged between the GSM module the host microcontroller using "AT Commands", a legacy computer language originally developed for dial-up modems. The module uses a mix of standard and propriety commands. AT commands always begin the "AT" characters and end with the carriage return character.

For example, the check the SIM card's own phone number, this command is sent:

```
AT+CNUM<cr>
```

AT commands and transmitted and their responses received on the asynchronous serial (UART) channel.

See the Telit Modules Software User Guide (http://www.telit.com/index.php?eID=tx_nawsecuredl&u=0&g=0&t=1409342439&hash=ce7639794a71937f8a1a095db7de3cde671e1057&file=downloadZone/522.pdf) for a complete list and explanation of AT commands.

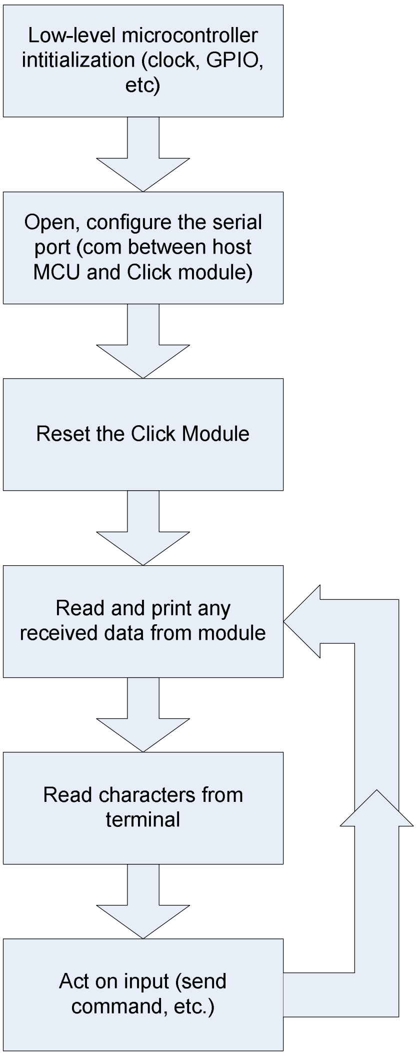
Example Module Configuration

The GSM module is very versatile, capable of sending data, voice, and other functions typical of cellular devices. The example software configures the module and sends an SMS (short message service) text message.

At initialization, the following AT commands are sent, in order. All commands in with the carriage return character (not shown in table).

Command	Description
AT	Wake-up
ATE0	Echo Off
AT+CMEE=2	Extended Error Codes Enabled
AT+CPIN?	Check SIM status
AT+CNUM	Display Phone number
AT#SLED=2	Enable Status LED (Should be the default!!)
AT+CREG=2	Set Registration Mode
AT+CREG?	Get Registration
AT+CMGF=1	Use ASCII Text format for SMS
AT+CNMI=2,2	Receive text messages directly via +CMT: unsolicited response
AT+COPS=4,0,T-Mobile	Select the operator as T-Mobile
AT+COPS?	Check Carrier/Registration
AT+CSQ	Get RSSI

Example Software Flow


Low-level microcontroller intitialization (clock, GPIO, etc)
Open, configure the serial port (com between host MCU and Click module)
Reset the Click Module
Read and print any received data from module
Read characters from terminal
Act on input (send command, etc.)

After initialization, users interact with the software through the console (standard input/output). Single character commands are entered which causes the software to send specific AT commands.

User Input	Action or AT Command
e	Sends escape sequence
h	Displays available commands
q	Quits the program
r	Check RSSI, AT+CSQ
s	Reads the service status, AT+COPS?
t	*Sends a text message. See note.
z	Resets the module

*Sending a text message requires entering the destination and phone number. Once entered, the AT commands sent are:

```
AT+CMGS=<phone number><cr>

<text message><Ctrl-Z>
```

Even if no SIM card is connected, the Click module should respond, possibly with an error code, to AT commands.

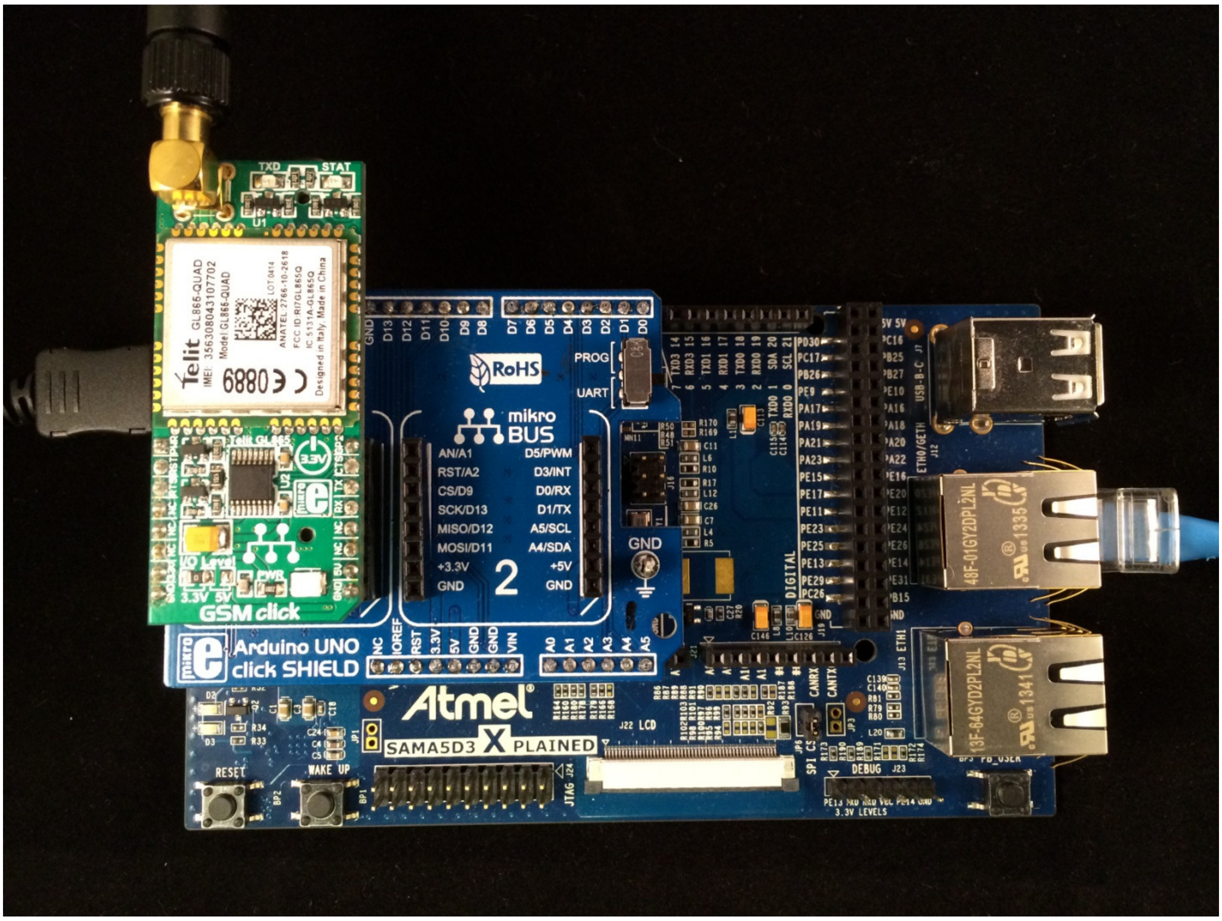
Connecting the GSM Click to the SAMA5D3 XPlained Board

Connecting the GSM Click board to the SAMA5D3 XPlained board requires using the Arduino UNO click shield adapter board (http://www.mikroe.com/click/arduino-uno-shield). The adapter board makes the following connections:

SAMA5 XPlained Pin	**SAMA5 XPlained Pin Configuration**	**Arduino UNO Click Shield Pin**	**GSM Click Pin**
3.3V		3.3V	3.3V
GND		GND	GND
PD23	Output	RST/A3	RST
PC29 / URXD0	Input	HDR TX	TX
PC30 / UTXD0	Output	HDR RX	RX
PC25	Input	PD2/INT0	SCL (acts as RTS)
PC9	Output	PB2/SPI-SS	SDA (acts as CTS)
PD23	Input (not used)	PC0/AN0	AN (acts as PWRMON)
PC18	Inputs (not used)	PD6/PWMA	PWM (acts as GPIO2)

The example code assumes that the Click Board is installed on the Arduino UNO Click Shield Adapter Bus 1. On the adapter board, SW1 must be set to "UART."

The example assumes that the SAMA5D3 Xplained board is running the factory-installed Poky Linux firmware.



In the default factory Linux configuration, the SAMA5 does not have the necessary power supply rail (VDDANA) enabled to use PDxx. This voltage supply is generated by an external regulator which is controlled by I2C commands. Before running the example program, enable the supply by executing the config_gpio.sh script which is part of the example program files.

```
# ./config_gpio.sh
```

Compile and Run the Example Program

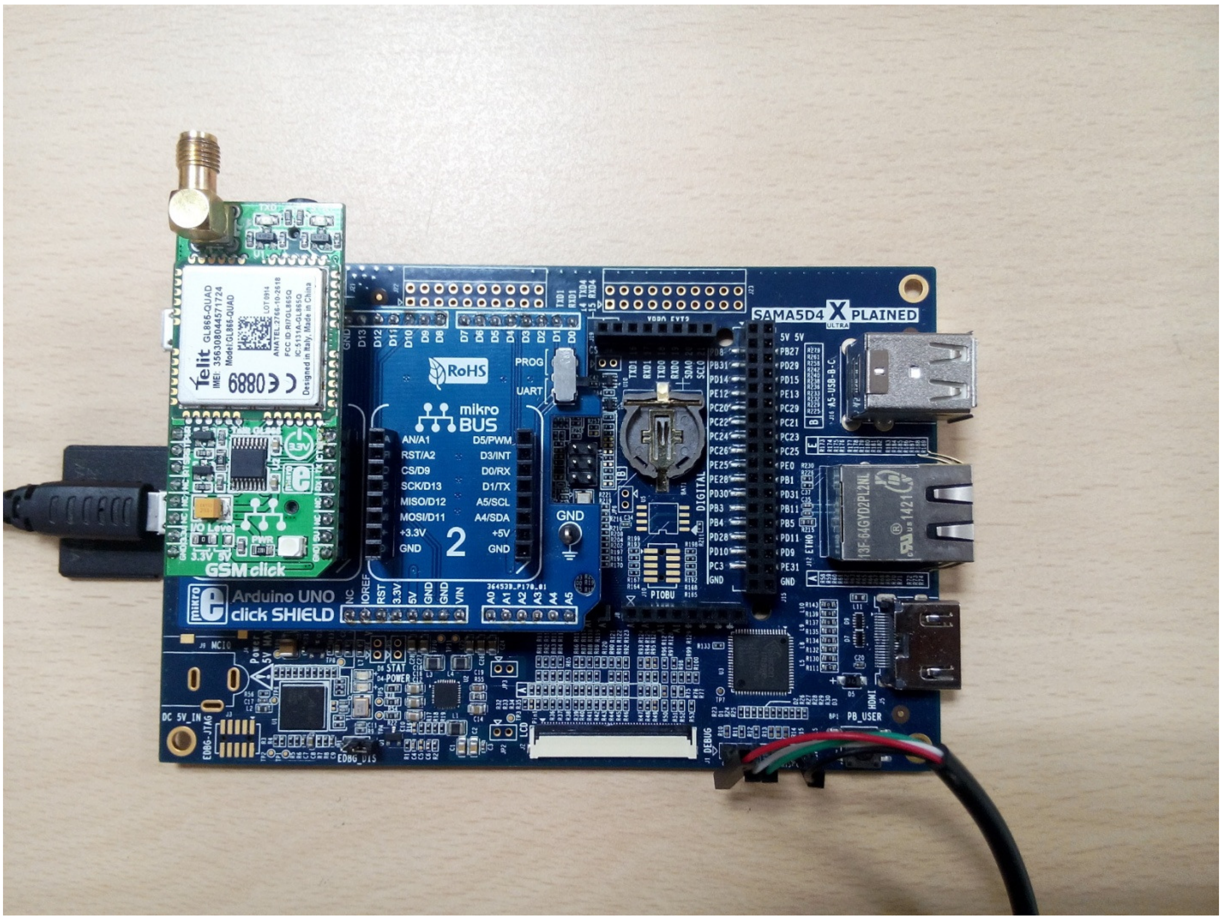
1. The example code requires a PC running Linux in order to cross-compile and install the program. The instructions for building the cross-compile toolchain can be found at http://www.at91.com/linux4sam/bin/view/Linux4SAM/PokyBuild#How_to_build_Poky_for_AT91. Build the toolchain.
2. Connect the SAMA5D3 XPlained board to an Ethernet connection and power-on the board. Using the serial port connection, connect to the SAMA5 board. From the shell, get the IP address of the SAMA5 board:
   ```
   # ifconfig eth0
   ```

3. Download the example code from XXXXXXXXXXXXXXXXXXXX and extract the files to convenient location.
4. Edit the makefile to use the actual IP address of the SAMA5 board.
5. Setup the build environment by running the following:
   # source /opt/poky/1.5.1/environment-setup-cortexa5hf-vfp-poky-linux-gnueabi
6. From the directory of the example program, compile the source code with make:
   # make sama5
7. Install the program into the target's file system at /home/root
   #make install_sama5
8. Run the program. In the SAMA5 shell, run
   # cd /home/root
   # ./ GSM_Click_sama5
9. Type single character commands (i.e. "r", "s") on the terminal followed by <enter>. The response from the module will be displayed.
10. To send a text message, enter the following, substituting 2165551212 with your destination's phone number:
    t<enter>
    2165551212<enter>
    Hello from SAMA5!<enter>

Connecting the GSM Click to the SAMA5D4 XPlained Board

Connecting the GSM Click board to the SAMA5D4 XPlained board requires using the Arduino UNO click shield adapter board (http://www.mikroe.com/click/arduino-uno-shield). The adapter board makes the following connections:

SAMA5 XPlained Pin	SAMA5 XPlained Pin Configuration	Arduino UNO Click Shield Pin	GSM Click Pin
3.3V		3.3V	3.3V
GND		GND	GND
PC30	Output	RST/A3	RST
PC25/ URXD1	Input	HDR TX	TX
PC26 / UTXD1	Output	HDR RX	RX
PB21	Input	PD2/INT0	SCL (acts as RTS)
PD10	Output	PB2/SPI-SS	SDA (acts as CTS)
PC30	Input (not used)	PC0/AN0	AN (acts as PWRMON)
PB28	Inputs (not used)	PD6/PWMA	PWM (acts as GPIO2)



The example code assumes that the Click Board is installed on the Arduino UNO Click Shield Adapter Bus 1. On the adapter board, SW1 must be set to "UART."

Compile and Run the Example Program

1. The example code requires a PC running Linux in order to cross-compile.
2. Download the toolchain from below link.

   https://releases.linaro.org/14.09/components/toolchain/binaries/gcc-linaro-arm-linux-gnueabihf-4.9-2014.09_linux.tar.bz2

   2.1 Extract the toolchain to home directories.

   ```
   $ tar -xvf arm-linux-gnueabihf-4.9-2014.09_linux.tar.bz2
   ```

   2.2 Write a shell script to create the toolchain.

   ```
   $ gedit toolchain.sh
   ```

```
export PATH=$PATH:/home/<user>/<path to bin directory of the downloaded toolchain>

export ARCH=arm

export CROSS_COMPILE=arm-linux-gnueabihf-
```

3. Download the example code from http://www.element14.com/community/docs/DOC-70203#cdownloads and extract the files to convenient location.

4. From within the directory of the example program, run the toolchain.sh and compile the source code by executing the below simple steps:

   ```
   $ . toolchain.sh
   $ arm-linux-gnueabihf-gcc -o “Executable name” “Source file.c”
   ```

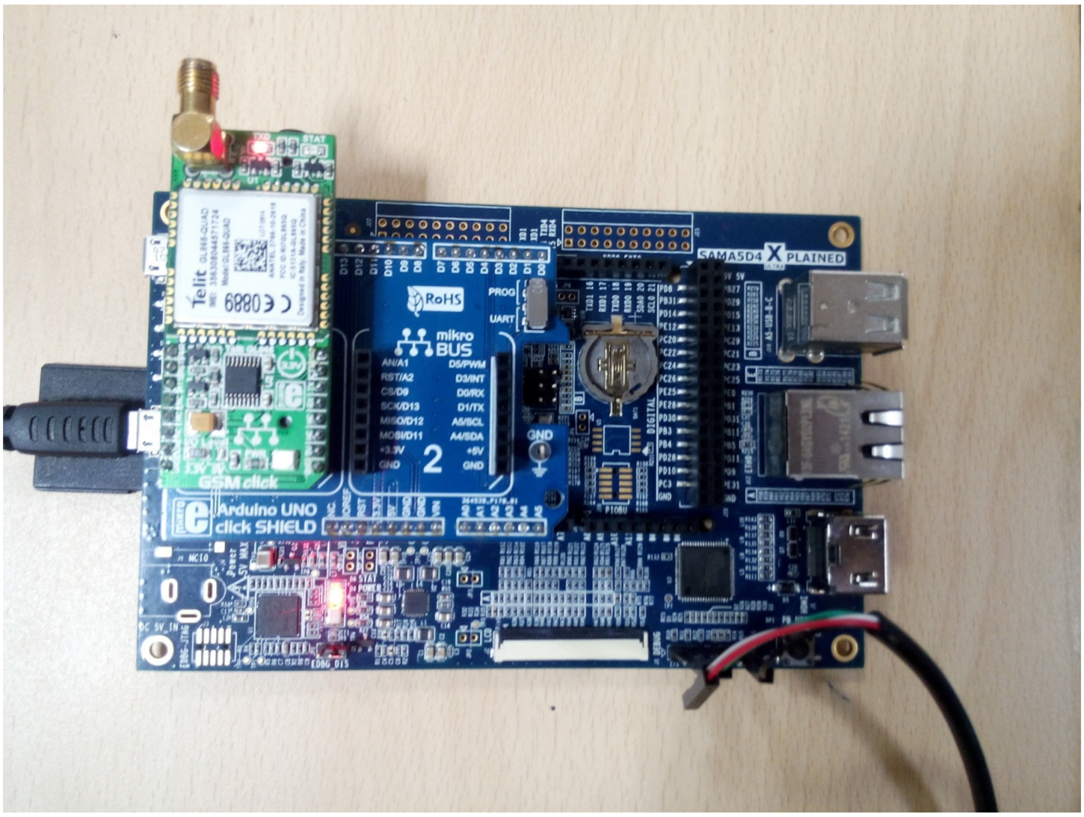
   Copy executable file into root partition of SD card

5. Connect the SAMA5D4 Xplained board to a PC using USB to serial debug cable and power-up the board by making appropriate jumper settings.

6. Run the program into SAMA5D4 target board

   ```
   root@arm:~# cd /
   ./Executable name
   ```

7. Type single character commands (i.e. "r", "s") on the terminal followed by <enter>. The response from the module will be displayed.
8. To send a text message, enter the following
   t<enter>
   Enter phone number: <enter>
   Enter Message:Hello from SAMA5D4!<enter>
   User should recevie sms on phone “Hello from SAMA5D4!”
9. Send sms “led” from your phone number to GSM module. After getting sms TXD LED of GSM module will switch ON as shown in below image.



Connecting the GSM Click to the BeagleBone Black

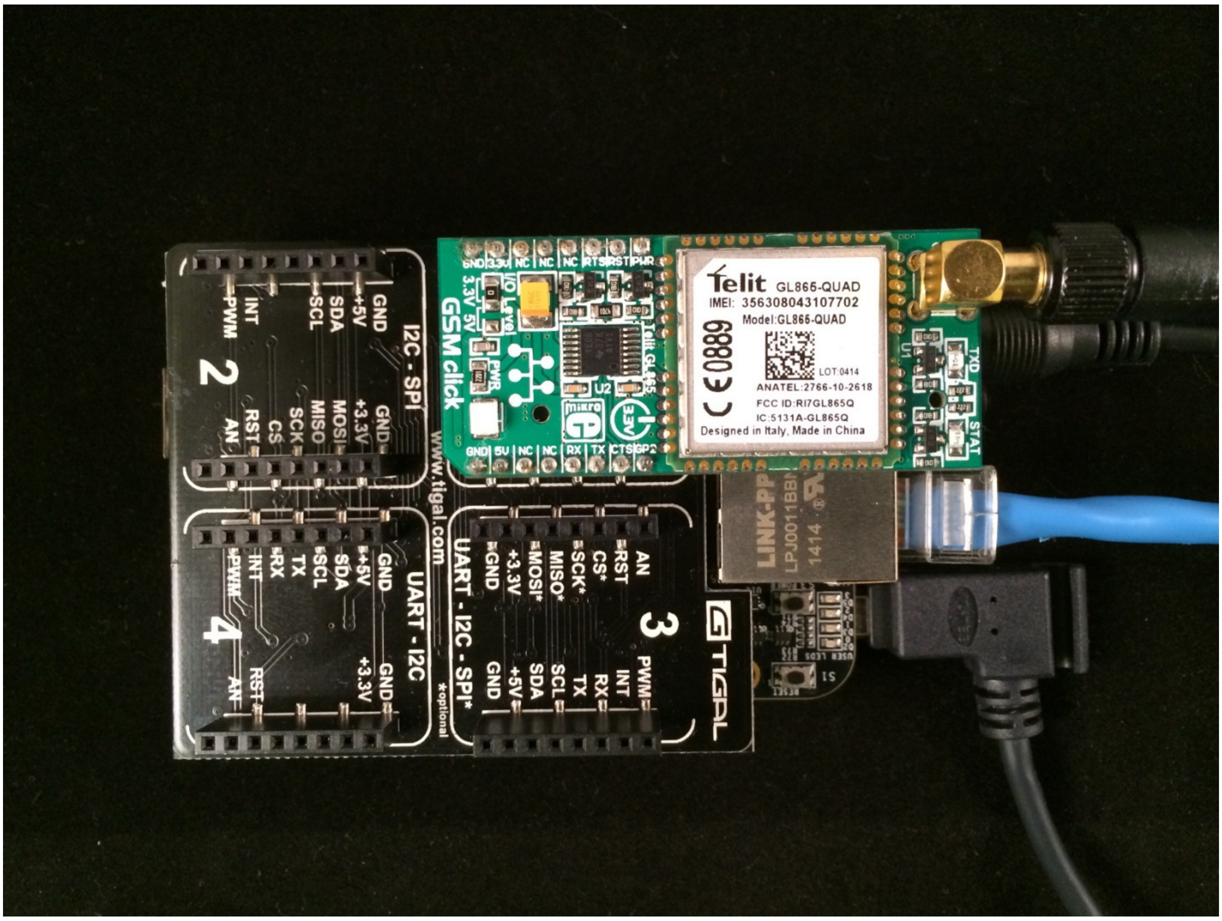
Connecting the GSM Click board to the Beaglebone Black requires using the Tigal BeagleBone mikroBUS Cape adapter board (http://elinux.org/Tigal:BeagleBone_mikrobuscape). The adapter board makes the following connections:

Beaglebone Black Pin	Beaglebone Black Pin Configuration	BeagleBone mikroBUS Cape Adapter Pin	GSM Click Pin
VDD_3V3EXP		3.3V	3.3V
Gnd		GND	GND
P9.12 (GPIO1-28)	Output	RST	RST
P9.21 (UART2_RXD)	Input	RX	TX
P9.22 (UART2_TXD)	Output	TX	RX
P9.28 (GPIO3-17)	Input	CS	SCL (acts as RTS)
P9.15 (GPIO1-16)	Output	INT	SDA (acts as CTS)
Not connected			AN (acts as PWRMON)
P9.14 (GPIO1-18)	Input (not used)	GP2	PWM (acts as GPIO2)

The example code assumes that the Click Board is installed on the BeagleBone mikroBUS Cape Adapter Host 1.

The example assumes that the Beaglebone Black is running the factory-installed Linux (debian) firmware.

Note CTS and RTS are on the I2C pins and changing them to GPIO will require modifying the base device tree, so the GSM_Click.c program does not use these pins.



Compile and Run the Example Program

1. Download the example code from XXXXXXXXXXXXX and extract the files to convenient location.
2. The example code requires gcc, BB-UART2-00A0.dtbo or BB-MIKROBUS-01-00A1.dtbo, and root access.
    a. Instructions to download BB-MIKROBUS-01-00A1.dtbo can be found at http://elinux.org/Tigal:BeagleBone_mikrobuscape
    b. https://www.tigal.com/wiki/doku.php?id=tigalcapes:bb_mikrobus_cape
        i. On this site they will also explain how to install the overlay and use it, along with other helpful information.
3. Next go to /sys/devices/bone_capemgr.9/
    a. This directory can differ depending on the OS you are running.
4. Enter in the following command echo BB-UART2 > slots
5. Now you can enter in the following command to see that BB-UART2 has been loaded correctly, cat slots.
    a. The output should look something like this:

```
0: 54:PF---
1: 55:PF---
2: 56:PF---
4: ff:P-O-- Bone-LT-eMMC-2G,00A0,Texas Instrument,BB-BONE-EMMC-2G
5: ff:P-O-- Bone-Black-HDMI,00A0,Texas Instrument,BB-BONELT-HDMI
6: ff:P-O-- Bone-Black-HDMIN,00A0,Texas Instrument,BB-BONELT-HDMIN
7: ff:P-O-L Override Board Name,00A0,Override Manuf,BB-UART2
```

   b. To remove this BB-UART2 enter in the following command -7 > slots.
   c. We enter in 7 since that is the number that BB_UART2 was assigned in our case.
      i. WARNING: The above command to unload the overlay may crash your system depending on the version you currently have installed.
      ii. The overlay will not be loaded when you reboot your system, unless you set it to load at boot at the /boot/uboot/uEnv.txt.

6. Compile the code using “gcc –Wall GSM_Click.c –o GSM_Click”
   a. –Wall enables all warnings
   b. –o Specify the output filename
   c. Technically you do not need either of these two flags to compile the sample code. They are just nice to have.
7. Run the program. In the BBB shell, run
   ./GSM_Click
8. Type single character commands (i.e. "r", "s") on the terminal followed by <enter>. The response from the module will be displayed.
9. To send a text message, enter the following, substituting 2165551212 with your destination's phone number:
   t<enter>
   2165551212<enter>
   Hello from BBB!<enter>

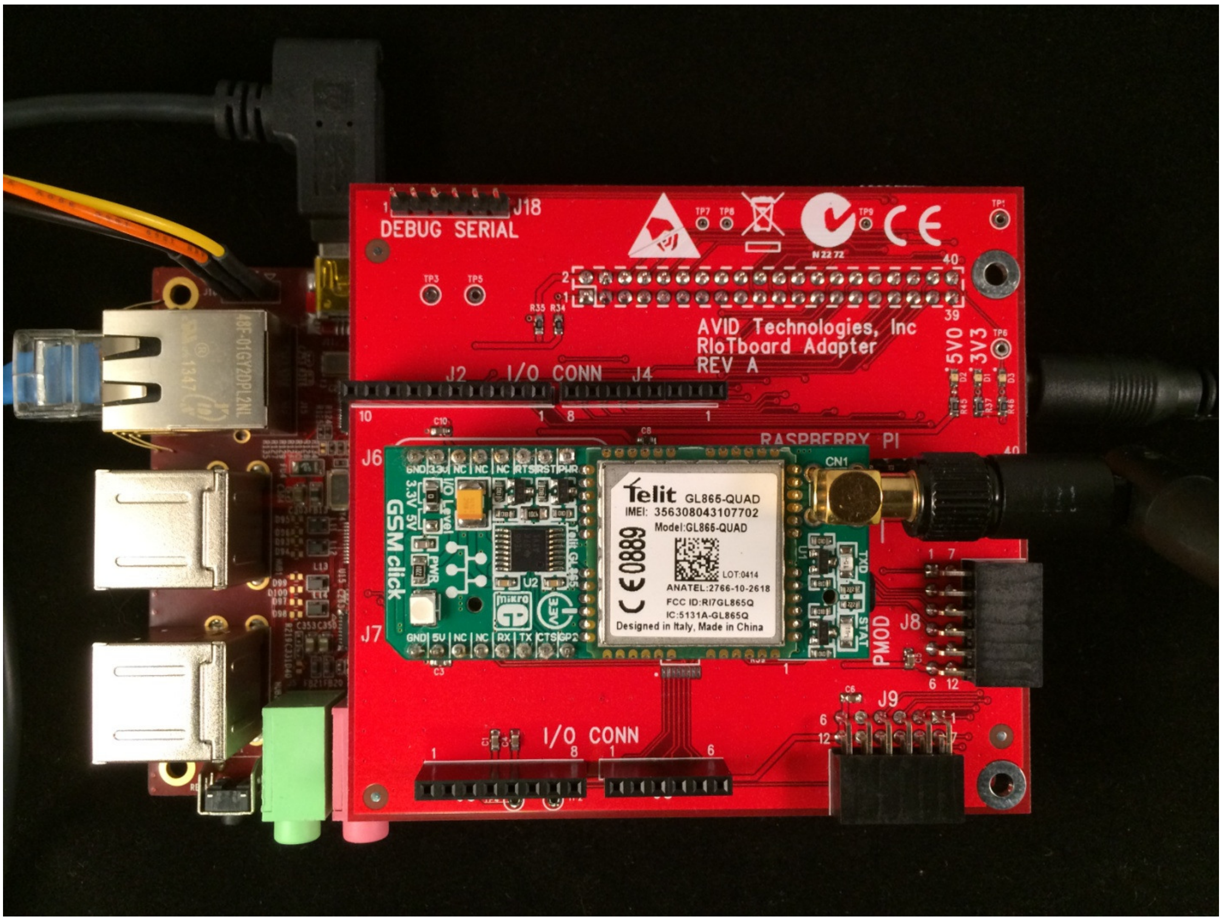
Connecting the GSM Click to the RIoTb Board

Connecting the GSM Click board to the RIoTboard requires the use of the RIoTboard Adapter board (http://www.element14.com/community/docs/DOC-68821). The adapter board makes the following connections:

RIoTboard Pin	RIoTboard Pin Configuration	RIoTboard Adapter Pin	GSM Click Pin
3.3V		3.3V	3.3V
GND		GND	GND
GPIO4_IO16	Output	CSPI2_CS0	RST
UART3_RX	Input	UART3_TXD	TX
UART3_TX	Output	UART3_RXD	RX
GPIO5_IO12	Output	I2C3_SCL	SCL (acts as RTS)
GPIO5_IO06	Input	I2C3_SDA	SDA (acts as CTS)

Not connected			AN (acts as PWRMON)
DISP0_DAT8	Input (not used)	PWM1	PWM (acts as GPIO2)

The example assumes that the RIoTboard is running the factory-installed Android firmware.



Compile and Run the Example Program

1. *The example code requires the installation of the following tools on the development PC:
    a. Android Software Development Kit (SDK)
    b. Android Native Development Kit (NDK)
    c. Java Development Kit (JDK)
    d. gcc and make tools (which can be installed by mingw MinGW)
    e. Cygwin
    f. A terminal emulator such as Putty.
2. Download the example code XXXXXXXXXXXX and extract the files to convenient location.
3. From the command line, change the jni directory in the example code.
4. Build the software with the ndk build tool:

```
> ndk-build
```

5. Install the software onto the RIoT Board:

```
> adb push libs/armeabi/ GSM_Click /data/local/tmp
```

6. Open a terminal connection (e.g. with Putty) to the target platform.
7. Change directories to the install location (/data/local/tmp)
   ```
   # cd /data/local/tmp
   ```
8. Make the file executable:
   ```
   # chmod 777 GSM_Click
   ```
9. Run the program
   ```
   ./GSM_Click
   ```
10. Type single character commands (i.e. "r", "s") on the terminal followed by <enter>. The response from the module will be displayed.
11. To send a text message, enter the following, substituting 2165551212 with your destination's phone number:
    ```
    t<enter>
    2165551212<enter>
    Hello from RIoTboard!<enter>
    ```