

Attila Mák 04.12.2017

## MOM

V Five Years Out





#### ©2017 ARROW

All rights reserved. No part of this manual shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical photocopying, desktop publishing, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of information contained herein. While every precaution has been taken in the preparation of this document, the publisher and author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein. All terms mentioned in this manual that are known to be trademarks or service marks are listed below. In addition, terms suspected of being trademarks or service marks have been appropriately capitalized. ARROW cannot attest to the accuracy of this information. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.





## **Revision History**

Revision, Date	Editor	Subject(major changes)
Revision 1.0,	Attila Mák	Initial release
04.12.2017	Attild Wak	lillidi i elease
Revision 2.0,	Attila Mák	Description changed
11.10.2018		

# Contents

1	Scop	pe/ description	5
2	Hard	dware setup	5
2	2.1	Arrow Quadro board	6
2	2.2	OMSEMI IDK board	7
3	Soft	ware setup	8
3	8.1	IDE installation to the PC	8
	3.1.	1 WICED installation	8
Set	up JRE	E 32 and JRE64 java runtime	8
Set	up Wi	iced Studio (WINDOWS)	9
SET	UP W	ICED STUDIO (OTHER OS)	13
C	DSX		13
L	inux 6	54-bit	13
L	inux 3	32 bit	13
	3.1.2	2 ONSEMI IDK installation	13
4	Usin	ng ONSEMI IDK	19
5	Qua	dro board	23

## 1 Scope/ description

This document is guiding to set up Arrow Quadro board with ONSEMI IoT kit. We will create a project which is an alarm system demonstration. If the PIR sensor detects any movement and the shutter is "closed" the alarm message will appear on the screen. And we will make a project with Quadro to send the received data to the cloud via wifi.

The connection between Quadro and ONSEMI IDK is based on UART.

## 2 Hardware setup

The required hardware to perform the steps described in this application note consists of:

**Developer PC:** 

This platform will be used for setup and writing and downloading the two firmware into the microcontrollers. The power supply task will be solved via USB. Desktop computer or laptop with x86 architecture and USB 2.0

The ARROW's Quadro board as it is

The ONSEMI IoT kit with LED driver board plus 12V / 2A Supply.

## 2.1 Arrow Quadro board

https://www.arrow.com/en/campaigns/cypress-wiced-iot



Arduino connection:

Х	SCL/D15
Х	SDA/D14
Х	SCK/D13
NOT CONNECTED	MISO/D12
IOREF	PWM/MOSI/D11
NRST	PWM/CS/D10
NRST	PWM/D9
+3.3V	D8
+5V	NOT CONNECTED
GND	NOT CONNECTED
GND	D7
+VIN	PWM/D6
AN0	PWM/D5
AN1	D4
AN2	PWM/D3
AN3	D2
AN4	UART_TXD_RXuc
AN5	UART_RX_TXuc

## 2.2 OMSEMI IDK board

http://www.onsemi.com/PowerSolutions/evalBoard.do?id=BB-GEVK



### Arduino connection:

	I2C_SCK1
	I2C_SDA
	AREF
	GND
	SPI_CLK1
3V3_KNX	SPI_DATAI1
IOREF	SPI_DATAO2
/RESET	SPI_CS
IOREF	SWO/DIO11
+5V	SWDIO/D13
VCOM	SWDCLK/D12
VCOM	SPI_DATAI2/DIO16
I2C_EXP_GPIO12	INT2
I2C_EXP_GPIO13	SPI_CLK2_DIO14
A3	PWM1
A2	INTO
A1	UART1_TX
A0	UART1_RX

## 3 Software setup

3.1 IDE installation to the PC

## 3.1.1 WICED installation

Pay ATTENTION! Some antivirus like BitDefender may cause problems during the installation phase and during normal use of the Wiced Studio

1) Download and install 32bit and 64bit JRE (Java Runtime Environment)

http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html

2) Download and install Cypress WICED Studio 4.1.1

https://community.cypress.com/community/wiced-wifi/wiced-wifi-documentation

3) Plug the Board with USB Cable

Verify Driver installation

4) Download and install Quadro BSP file

BCM943907\_QUADRO\_w6.zip for WICED 6.0 IDE

5) Download the Quadro IoT Starter Kit Getting Started Guide

Bluemix IoT LAB

### Setup JRE 32 and JRE64 java runtime

- 1) Check/Fix your JRE (Java Runtime Environment) installation:
  - a. 32-bit JRE is needed for Cypress WICED Studio
  - b. 64-bit JRE is needed for SDK's installer
    - (JRE is designed to allow both 32 and 64 bit variants to be installed on same system)
- 2) Not normally required, but if you have a JRE related issue, check your Windows PATH. This should include a path to your Java installation:

C:\ProgramData\Oracle\Java\javapath





#### Setup Wiced Studio (WINDOWS)

- 1) install Cypress WICED Studio 4.1.1 development tools by unzip file WICED-Studio-4.1.1.8-IDE-Installer.exe.zip and execute WICED-Studio-4.1.1.8-IDE-Installer.exe
- 2) When Setup start Appear window below

InstallAnywh	ere	
WIGED	Installazione di InstallAnywhere in preparazione	
	20%	
		Annula

#### And then



#### Press Next Button



Press Next Button





#### Press Next Button



#### Press Install



Wait while setup finish





#### And press Done Button



Select 43xxx\_Wi-Fi Platform

£	×
Select WICED Platform	
Select the default WICED Platform you intend to use for your development. You can change the selection at any time from the Eclipse IDE 'WICED Platform' drop down menu after initial selection.	
○ 20706-A2_Bluetooth	
○ 20719-B0_Bluetooth	
○ 20729-B0_ZigBee	
O 20735-B0_Bluetooth	
④ 43xxx_Wi-Fi	
ОК	





## The WICED IDE must appear as below

C/C++\README.txt - Eclipse				– a ×
File Edit Source Refactor N	avigate Search Project Run WICED Platform Window Help			
	n 🗟 🔌 🕩 🗉 🔳 💦 🕉 🕫 🕫 😴 43xxx_Wi-Fi 🛛 🗸 🖬 😭 🕶 🗳 🕶 🗳 🗸 🖓	s • 🜔 • 🤮 • 💁 •	😂 😂 🛷 👻 🖢 😴	2 · · · · · · ·
				Quick Access
Project Expl 22 C	READMEtxt S3 2 Cypress MICED Studio Software Development Kit 4.1 - README 3 Cypress MICED Studio Software Development Kit 4.1 - README 4 5 MICED Studio provides systems and APIs needed to build, design and implement 6 applications for wil-Fi, Bluetooth Classic (BR/EDR), Bluetooth low energy (BLE), 7 and ZigBee devices. 8 9 MICED Studio platforms include support for - 10 - 20706-42, 20735-8 and 20719-90 based Bluetooth platforms. 11 - Support for various Cypress Mi-Fi & combo chips 2 - 43908 (43904, 43304) Mi-Fi SoC 3 - 43308 (43304, 43304) Mi-Fi SoC 13 - 43348 (43434, 4334) Mi-Fi SoC 15 - 43340 Mi-Fi + Bluetooth combo SoC 15 - 2073-80 based Ziefee nlatforms. 10 - 2070-80 based Ziefee nlatforms. 11 - 2000 - 2070-2000 - 200		<ul> <li>Make Target ⊠</li> <li>➢ 43∞x_Wi-Fi</li> <li>Make Target 20</li> <li>Make Target 20</li> </ul>	
	<ol> <li>20/25-06 Gased Zigbee Pictforms.</li> <li>17</li> <li>18 Bluetooth Features:</li> <li>19 - API to access Bluetooth stack including GAP, ATT, GATT, SMP, L2CAP,</li> <li>20 RFCORM, SDP, A2DP and AVRC in the ROM.</li> <li>21 - A generic profile level API that abstracts the Bluetooth stack layer API.</li> <li>22 - Reference applications for profiles defined by the Bluetooth SIG.</li> </ol>	>	Contents % Searce Index Cypress WICED Studio	th 📽 Related Topics 🕼 Bookmarks
	Caracter M D Backlana & Caracter Ma Datura		WICED S	tudio™
	No consoles to disolav at this time.		N CONESS.	
			Development This is a very brief intr Development System the formal WICED Stur Building & Downlo	ent System oduction to the WICED Studio for those reckless souls that refuse to read dio Quickstart Guide ading Applications
			Plug the WICEE	Studio Evaluation Board into the

## Expand 43xxx\_Wi-Fi Project Explorer and 43xxx\_Wi-Fi Make Target

#### 📀 C/C++ - .\README.txt - Eclipse

File Edit Source Refactor N	lavigate Search Project Run WICED Platform Window Help						
📑 🕶 🗄 🕒 🕒 👻 🔨 🕶	r 🗟 🔌 🕨 🗉 🔳 🕅 3. 🐵 . R 🔜 😿 43xxx_Wi-Fi 💦 🗸 🗐 🕤 🔂 🕶 🗳 🕶 🧭	• 🔅 • 🜔 • 💡 • '	<b>9</b> - I	😂 😂 🛷 🔹 🐓 👻 🖛	$\leftarrow \diamond \bullet \bullet$		
					Quick Access	EC/C+	+
🎦 Project Expl 🔀 🖳 🗖	README.txt 🕱	-		Make Target      S		> 📐 🗆 C	5
<ul> <li>Construction</li> <li>Construction</li></ul>	<pre>1</pre>	> * • • • •	•	<ul> <li>Asoc Wi-Fi</li> <li>Gean</li> <li>Geom appliance-Bi</li> <li>Gemo appliance-Bi</li> <li>Gemo appliance-Bi</li> <li>Sinjuscan-BCM943</li> <li>Sinjuscan-BCM9444</li> <li>Sinjuscan-BC</li></ul>	CM943362WCD4 download I-BCM943362WCD4 download I-BCM943362WCD4 download S62WCD4 download S62WCD4	i run load run d run d sDIO downla sDIO downla marks use to read	
	W	/ritable Insert		6:45			

MOM

V Five Years Out

o ×

### SETUP WICED STUDIO (OTHER OS)

#### OSX

https://community.cypress.com/docs/DOC-3988

### Linux 64-bit

https://community.cypress.com/docs/DOC-3989

### Linux 32 bit

https://community.cypress.com/docs/DOC-3990

#### 3.1.2 ONSEMI IDK installation

## Java Installation JRE/JDK version 8u101 or above needs to be installed on the PC: http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

teva SE	Overview D	ownloads	Decumentatio	on Com	munity	Technologies	Training
lava EE							
tava ME	Java SE	Runtim	- Environ	mont R	Down	sheele	
lava St- Superat	Davarment	to run Java		to you wan	to daws	for Java program	no? If you want to run
Inva RE Advanced & Suite	Java program	rs, but not d	evelop them, do	wricad the	Java Ru	ntime Environme	ent, or JRETN
ava Embedded	H you want to	develop an	olications for la	a countra	d the la	va Doueleomon	Kit or IDKIN The ID
and binesses	inch des the	IRE sown	do not have to d	ew lose to	sh eans	rateb.	ALCOLOGIC THE SEA
ava DB	incides incides	ME, 50 500	ab normate to a	CWI ICED LY	Jui sepa	ately.	
//eb Tier	JRE Bu101 C	hecksum					
lava Card	SHE OUTUR O	necksum					
lava TV		Ja	va SE Run	time E	nviro	nment 8u1	01
vew to laws	You must	accept the	Oracle Binary C	ode Licen	se Agre	ement for Java	SE to download this
horn all all the		0.		SOILWA			
Johnnanny		U A	coept Libense #	greement	ωυ	ebine Lieense	Agreement
lava Magazine	Produc	t/ File Dead	cription	File Size		Dow	nicad
	LINEX XSS			54 C2 N/B	Jan 6	un Dr. Januar-1586	rpini
	Linux xoo			C3105 MD	10-0	of 01 January College	181 (42
	LINCK NOV			CO JONIC	pe-o	and the second dis-	201
	LINLX X04			68 40 MB	100-8	un un -anux-x64.6	ar gz
	Mac OS X			COTOME	90-8	unun -maccax-xt	4 tar gz
	Mac OS X			64 32 MB	J10-8	u101-maccax-x6	4 drag
	Solaris SPW	RC 64 5k		52 MB	jel	u101 sobrie spa	ir sv9 tar gz
	Solans x04			49 85 MB	10-8	uri Bri-solants-x64	tar gz
	Windows x80	3 Coline		071NB	10-8	ut 01-windows-it	580-illw cac
	Windows x88	S Cffine		52 63 MB	jre-8	ut 01 -windowa-i8	ERE EXE
	Windows sPF	3		59.42 MR	jie 8	utilit windows is	iAR tar gz
	Windows x64	4 Cffine		59 17 MB	00-8	ut 01-windows-x	64 exe
	Windows x64	1		62 77 MB	jin-fl	ut01-windows-a	64 tar g2
	You anize	Ja	va SE Run	time E	nviro	nment 8u1	02
	Too must	accept the	oracle binary (	softwa	ire.	ensem for sava	ac to download this
		O A	ccept License A	greement	• 0	ecline License	Agreement
	Burden	I Plie Press	and and in co.	Pile Pier		Barro	aland.

### GNUToolchain

The GNU cross compiler needs to be installed to compile the IDK application. Double click on the GNUToolchain.exe to install the cross compiler. Internet connection is mandatory to install the cross compiler.



Name	Date modified	Туре	Size
Gnutoolchain.exe		Application	163 KB
IDK_Installer_x86.exe		Application	145,726 KB
IDK_Installer_x86_64.exe		Application	145,854 KB

Select the "GNU Toolchain" checkbox and click Next.

Choose which featur	res of GNU Toolchain you wa	nt to install.
Check the componer nstall. Click Next to	nts you want to install and un continue.	ncheck the components you don't want to
Select components t	to install: GNU Toolch	Position your mouse over a component to see its description.
Space required: 420	.OMB	

Select Destination folder and click Next. It is recommended to not change installation path.

Choose Install	Location			6
Choose the fold	ler in which to install GNU	Toolchain.		
Setup will instal	GNU Toolchain in the foll	owing folder. To install	in a different	folder, click
Browse and sei	ect another folder. Click I	nstall to start the insta	llation.	
Destination F	older			
Destination F	older 1008129\Documents\GNU	JToolchain	В	owse
Destination F	older 1008129\Documents\GNL : 420.0MB	(Toolchain	B	owse
Destination F	older 1008129\Documents\GNU : 420.0MB : 3.9GB	JTookhain	В	owse
Destination F <b>ENUSCION</b> Space required Space available	older 1008129/Documents/GNU : 420.0MB : 3.9GB	JToolthain	B	owse

Installer automatically downloads toolchain and installs.

8	GNU Toolchain Setup	×
<b>Installing</b> Please wait whi	le GNU Toolchain is being installed.	
Downloading To	polchain.zip	
Show details	S	
	Connecting	
	Cancel	
	< <u>B</u> ack <u>N</u> ext >	Cancel



GNU Tool chain installation	complete.
8	GNU Toolchain Setup -
	Completing GNU Toolchain Setup
	GNU Toolchain has been installed on your computer.
	Click Finish to close Setup.
PAN	
1 Andrews	
	< Back Einish Cancel

**IDK** Installation

Double click on the installer downloaded from ON Semiconductor. For 32 bit machines, install IDK Installer x86.exe. For 64 bit machines, install IDK Installer x86 64.exe

Name	
IDK_Installer_x86.exe IDK_Installer_x86_64.exe	32 bit architecture 64 Bit architecture

V

Read the license, check the box and click Next.

License Agreemen	nt		
Please review the li	icense terms before installing IDK.		
Press Page Down b	o see the rest of the agreement.		
ON Semiconductor	Software Development Tools End User Licens	se Agreement ("EULA")	^
This Software Dev entered into betwe limited liability com McDowell Road, PH ("Licensee"), eithe that accompanies firmware, associat ("Software"). YOU	elopment Tools End User License Agreement een SEMICONDUCTOR COMPONENTS INDUST pany and its subsidiaries and affiliates with ar noenix, Arizona 85008, (dba "ON Semiconduc r an individual or a single entity for the ON Se this EULA, which includes computer software ted media, printed materials, and/or electronic AGREE TO BE BOUND BY THE TERMS OF THI	("EULA") is made and TRIES LLC, a Delaware address at 5005 E. tor") and you emiconductor software and may include c documentation S EULA BY INSTALLING,	*
If you accept the to agreement to insta	erms of the agreement, click the check box be II IDK. Click Next to continue.	elow. You must accept the	
I accept the ter	ms of the License Agreement		

Choose the destination directory to install the IDK. It is recommended to have IDK installed under

C:\OnSemiconductor or D:\OnSemiconductor.

If a previous workspace is being retained, then make sure that metadata folder inside Workspace directory is deleted.

l.	IDK S	Setup			×
Choose Install Lo	cation				
Choose the folder	n which to install IDK.				
Setup will install ID select another fold	K in the following folder. To er. Click Next to continue.	install in a differ	ent folder, clic	k Browse and	
Destination Fold	r				
Destination Fold	s ductor		E	jowse	
Destination Fold	s ductor		E	gowse	
Destination Fold D:\OnSemicon Space required: 22 Space available: 5	r ductor 0.6MB 3GB		E	jowse	
Destination Fold D:\OnSemicon Space required: 22 Space available: 5	r ductor 0.6MB 3GB		E	gowse	
Destination Fold DE/OnSemicon Space required: 22 Space available: 5	r Evictor 0.6MB 3GB	< Back	E	Browse	

	IDK Setup	- 🗆 🗙
Installation Compl Setup was completed	ete ed successfully.	
Completed		
Show <u>d</u> etails		
	< Back	Cancel

MOM

Once in is successfully installed, a shortcut will be created on the desktop. Double click on the IDK shortcut on the desktop to launch the IDK IDE. The ON Semiconductor splash screen will launch, followed by the Welcome Screen.

		C/C++ = EDK	
Die Lan	Source Refector Severate Search Bropert Exe	mples Bus Window Lleip	
e 98 V	Vekone-12		D + + * * 13 - 4
<b>n</b>			and the second s
	Welcome to the	OnSemi IOT Development E	invironment
	First Steps     Take your first steps	Overview Get an exercise of the feature	
	Lutorials Co through tatarian	Samples by out the samples	

## 4 Using ONSEMI IDK



Then select the desired workspace name



Here the example Workspace name is the default.



```
//Initialize Serial instance
Serial screen(USBTX,USBRX);
Serial Quadro(p0,p1);
// create a PIR class object.
```

```
// create a PIR class object.
NCS36000 pir;
// create a LCD class object.
NHD_C0216CZ lcd;
//butons
IOXP_BTN bb keys;
//motor driver panel
AMIS30543D stepper2(MOTOR2);
```

```
#define INT_SET 1
#define INT_CLEAR 0
//variables
unsigned int button_status = BUTTON_INVALID;
bool buttonPress = BUTTON_FALSE;
bool alarmState = false;
bool TR_sensed = false;
bool ToggleSwitch = false, Closed = false;
```

#### Includes, variables for the project



#### Exit demo routine

```
void button routine (uint8 t button)
₽ (
        int ret=0;
       if ( button == BUTTON_LEFT )
       £
            lcd.displayString("LEFT");
            Quadro.printf("LEFT");
       else if (button == BUTTON_DOWN)
       ł
            lcd.displayString("DOWN");
            Quadro.printf("DOWN");
            if(!Closed)
            ł
                 ret = stepper2.rotateAngle(-1440);
                 if(ret!= STPR RET_SUCCESS){
    screen.printf("motor driver 2 : rotating motor -1440 degrees failed, error : %d !!!\r\n", ret);
    lcd.displayString("MOTOR1:ROTATION\n-1440 Deg failed");
                      wait(1);
                      lcd.displayString("ERROR !!!\nEXITING DEMO");
exit_demo();
                 }else{
                      screen.printf("motor driver 2 : rotating motor -1440 degrees success ...\r\n");
lcd.displayString("MOTOR1:ROTATION\n-1440 Deg OK");
                 Closed = true;
                 wait(1);
```



```
else if (button == BUTTON OK)
     ł
          if(ToggleSwitch==true)
              ToggleSwitch=false;
              lcd.displayString("Disalarmed");
              Quadro.printf("Disalarmed");
          }
          else
          £
              ToggleSwitch=true;
              lcd.displayString("Alarmed");
              Quadro.printf("Alarmed");
     else if (button == BUTTON_UP)
     ł
         lcd.displayString("UP");
Quadro.printf("UP");
          if(Closed)
          ł
              ret = stepper2.rotateAngle(1440);
if(ret != STPR_RET_SUCCESS){
    screen.printf("motor driver 2 : rotating motor 1440 degrees failed, error : %d !!!\r\n", ret);
                             lcd.displayString("MOTOR1:ROTATION\n1440 Deg failed");
                             wait(1);
                             lcd.displayString("ERROR !!!\nEXITING DEMO");
                             exit_demo();
                        }else{
                             screen.printf("motor driver 2 : rotating motor -1440 degrees success ...\r\n");
lcd.displayString("MOTOR1:ROTATION\n1440 Deg OK");
                        Closed=false;
                        wait(1);
      else if (button == BUTTON RIGHT)
           lcd.displayString("RIGHT");
           Quadro.printf("RIGHT");
      'n
      else
      ł
           lcd.displayString("Invalid");
      3
  }
  void pir_routine()
Ę
  ł
       if((ToggleSwitch==true)&&(Closed))
       ł
            lcd.displayString("Movement Detected");
            PIR sensed=true;
screen.printf("Movement Detected\r\n");
Quadro.printf("Move\n");
       }
  }
```

How to handling the buttons on the board



```
void pir_routine()

{
    if((ToggleSwitch==true)&&(Closed))
    {
        lcd.displayString("Movement Detected");
        PIR_sensed=true;
        screen.printf("Movement Detected\r\n");
        Quadro.printf("Move\n");
    }
}
```

#### **PIR sensor routine**

```
]int main()
Ξŧ
        int ret=0;
       lcd.init():
       lcd.displayString("ALARM demo with QUADRO ");
       wait(2):
       pir.registerCallback(pir_routine);
lcd.displayString("PIR initialized");
       wait(2);
       ret = stepper2.enable();
if(ret != STER_RET_SUCCESS){
    screen.printf("motor driver 2 : enabling the stepper motor 2 failed, error : %d !!!\r\n", ret);
    lcd.displayString("MOTOR2:\nEnable Failed");
             wait(1);
             lcd.displayString("ERROR !!!\nEXITING DEMO");
              return (-1);
       }else{
             screen.printf("motor driver 2 : enabling the stepper motor 2 success ...\r\n");
lcd.displayString("MOTOR2:\nEnable success");
       wait(1);
       //This API should be called in the while loop, if any, in the main application during each iteration of loop.
//Should be called at least once in an application without loop.It should not be called inside any ISR.
//Clears any interrupt caused by ERR out pin in stepper shield gpio expander.
ret = stepper2.checkStprErrorOut();
                    if(ret == true){
                          screen.printf("ERR OUT set low, check status registers for exact reason of error !!!\r\n");
lcd.displayString("MOTOR2:\nERR OUT LOW");
                    }else{
                          screen.printf("ERR OUT set HIGH, No errors ...\r\n");
lcd.displayString("MOTOR2:\nNO ERR OUT");
                    //enable motor2 peak coil current as 715mA.
ret = stepper2.setMcPeakCurr(9);
                    if(ret != STER RET_SUCCESS) {
    screen.printf("motor driver 2 : setting the peak current of motor 2 coil to 715mA failed, error : %d !!!\r\n", ret);
    lcd.displayString("MOTOR2: Coil\nPeak cur not set");
                           wait(1);
                            lcd.displayString("ERROR !!!\nEXITING DEMO");
                           return (-1);
                    }else{
```

screen.printf("motor driver 2 : setting the peak current of motor 2 coil to 715mA success ...\r\n"); lcd.displayString("MOTOR2: Coil\nPeak cur set"); .
wait(1); //set the desired stepper mode rst = stopper2.setStepperMode(MICRO\_1\_4);
if(ret != STFR\_RET\_SUCCESS) {
 screen.printf("motor driver 2 : setting stepper mode to micro step 1/4 failed, error : %d !!!\r\n", ret); lcd.displayString("MOTOR2:MICRO1/4\nMODE not set"); wait(1); lcd.displayString("ERROR !!!\nEXITING DEMO"); return (-1); }else{ screen.printf("motor driver 2 : setting stepper mode to micro step 1/4 success ...\r\n"); lcd.displayString("MOTOR2:MICRO1/4\nMODE set"); wait(1); /\*Initialize the button library\*/ bb\_keys.init(); /\*Register button/key events\*/ bb\_keys.registerCallback(button\_routine); while(1) lcd.clearDisplay(); wait(3);



}



This is the main function.

# 5 Using Quadro board

Here we use the email sample /snip program. Just we add there some functions.

101		
102	wiced_result_t	<pre>ret = WICED_SUCCESS;</pre>
103	wiced_result_t	result;
104		
105	wiced_ring_buffer_t	<pre>rx_buffer;</pre>
106	uint8_t	<pre>rx_data[RX_BUFF];</pre>
107		
108	wiced_ring_buffer_t	Q_rx_buffer;
109	uint8_t	<pre>Q_rx_data[RX_BUFF];</pre>
110		
1116	⊜ /****************	******
112	*	Structures
113	*****	***************************************
114	wiced_uart_config_t	uart_handle =
115	{	
116	.baud_rate =	115200,
117	.data_width =	DATA_WIDTH_8BIT,
118	.parity =	NO_PARITY,
119	.stop_bits =	STOP_BITS_1,
120	.flow_control =	FLOW_CONTROL_DISABLED,
121	};	
122		

Structs for serial communication.

1409	<pre>void application_start(</pre>	)
141	{	
142	uint32_t	<pre>expected_data_size = 1;</pre>
143	int	retries = 0,i=0;
144	int	<pre>count = 0, cnt=0;</pre>
145	uint8_t	incomingDATA[RX_BUFF];
146	uint8_t	outDATA[RX_BUFF];
147	char	newmsg[1024];
148	char c;	
149		
150		
151	<pre>/* Initialise WICED</pre>	device */
152	<pre>wiced_init( );</pre>	

Inside the application\_start function the variables what we need. And the wiced init.

```
153 ring_buffer_init(&rx_buffer, rx_data, RX_BUFF );
154 ring_buffer_init(&Q_rx_buffer, Q_rx_data, RX_BUFF );
155
156 /* Initialize <u>Uart</u> */
157 wiced_uart_init(<u>WICED UART 2</u>, &uart_handle, &rx_buffer );
158 wiced_uart_init(<u>WICED UART 1</u>, &uart_handle, &rx_buffer );
159
```



This is how its looks when working:

