LAPIS PC Application

User's Manual

Document Number: TBD

Rohm Semiconductor LLC

Santa Clara, California USA

Table of Contents

Table of Figures

Revision History

Version	Description	Date	Initials
0.1	Initial Creation of Document	06/17/2013	KB
0.2	Finalized Edits	6/28/2013	KB/JC
0.3	Added Sections for I2C and Q111 usage	10/17/2013	KB

1. Introduction

1.1 Purpose

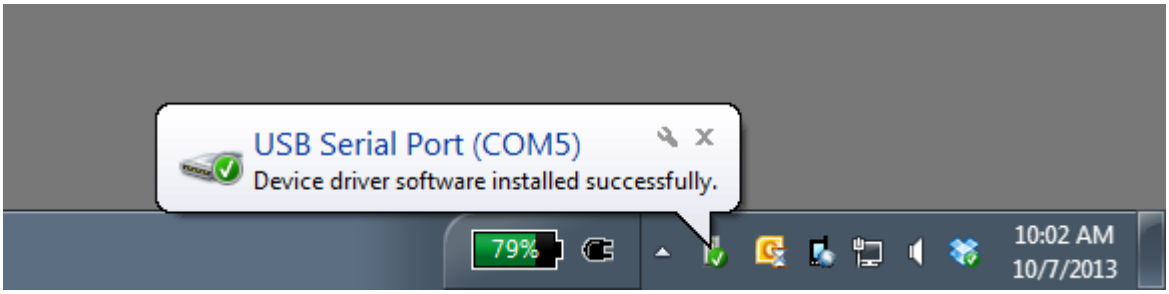
This user's manual provides the basic information necessary for a first time user to use the PC application that is used to control the default firmware on the LAPIS development platform.

1.2 PC/System Requirements

This PC Application was designed for Windows Based PCs. This software has been developed for Windows 7 and Windows 8 and supports 32bit and 64bit architecture. In order to be sure that the COM port FTDI driver on the LAPIS board is loaded properly, please be sure the PC has the latest windows updates and can connect to the internet.

1.3 Driver Requirements

When the LAPIS is connected directly to the PC; windows hardware wizard should handle finding the appropriate driver for the LAPIS device. Please note the COM port that is connected on your PC. The COM5 shown below is only an example; this port may be different depending on the user's setup.



Manual Download for FTDI Drivers: http://www.ftdichip.com/FTDrivers.htm

2. PC Application

2.1 GUI Interface

The Graphical User Interface (GUI) for the PC application is designed using TABS embedded within the main window. Each tab has a different function implemented in order to access different functionalities within the LAPIS MCU. For example, the GPIO Tab can be used to configure the real-time GPIO settings of the Q112 MCU.

Please note that the other MCU (Q111) on this board does not currently work with this PC application interface. This feature will be added in a later revision.

The PC Application can run without installation by double clicking the **LAPIS_ControlSoftware.exe** file. As the driver is connected via FTDI Com Port driver, this application is essentially a COM port application.

Figure 1: Main Application Screen

2.1.1 Config/UART Tab

Once the user opens up the PC application, Tab 1 (Config/UART) will be opened as default. This tab allows the user to connect the PC GUI to the User's application.

Please note that in this PC application, the UART connection is used as the communication protocol for this GUI application's other functions. Thus, we must be sure that this tab is connected before other tabs are used.

In order to ensure connection, the PC application software will not allow the user to change tabs until a port is opened. Also, the connection indicator will also change from "Not Connected" to "Connected" when the PC software and UART connection have been successfully established.

Figure 2. Below shows the main blocks of this tab. The left side allows the user to connect the PC software to the appropriate COM Port. The right side allows the user to check to see if the UART connection is established properly.

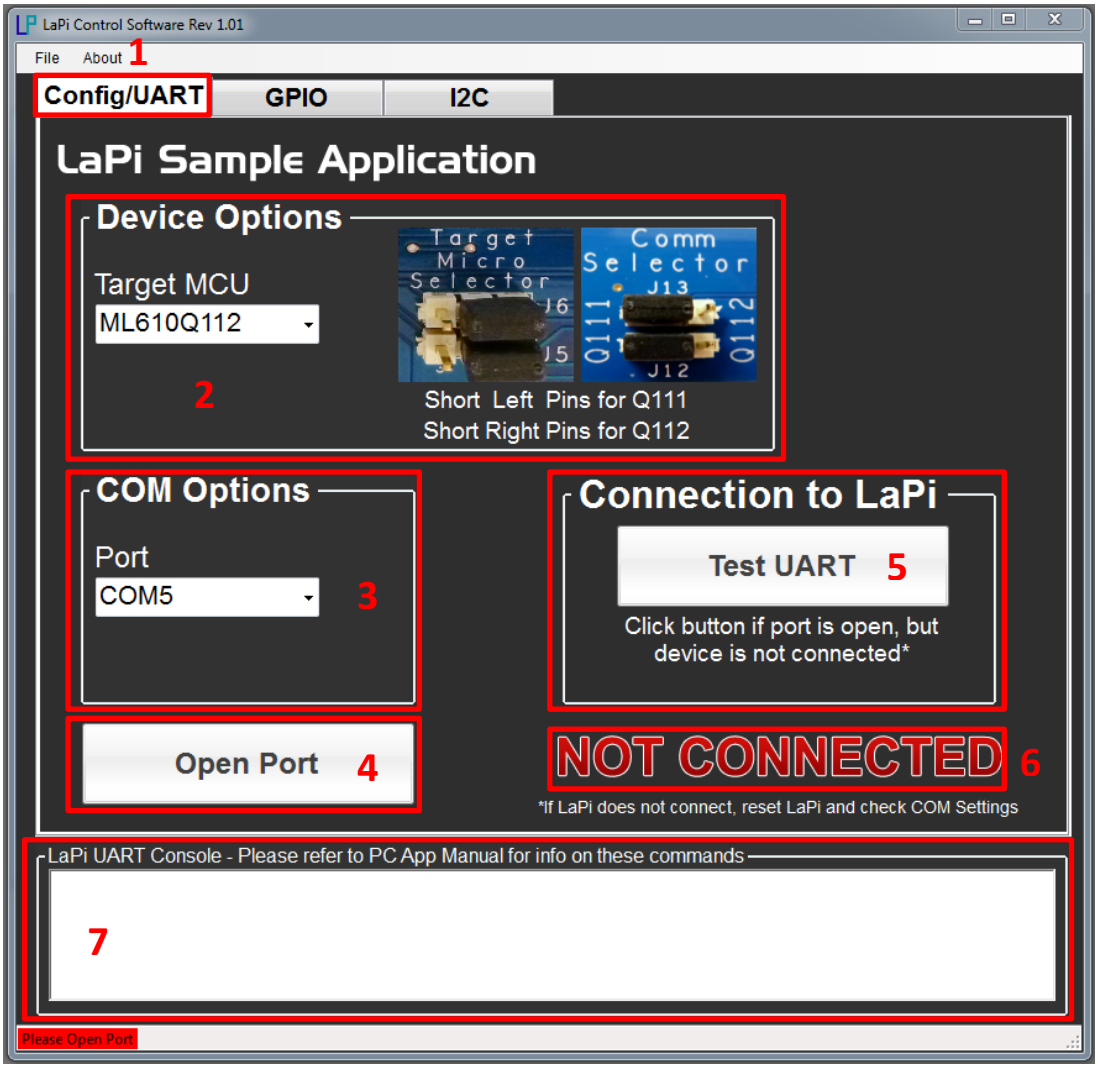


Figure 2: Config/UART Tab

2.1.2 *Tab Information*

1. Config/UART Tab:

Clicking on the "Config/UART" tab will bring the user back to this screen.

2. Target MCU

Please indicate the Target MCU you would like to evaluate. Additionally, you will need to adjust the "COMM Selector" and "Target Micro Selector" jumpers based on which device you would like to use.

- Short both sets of jumpers to the **LEFT** for operation of the **ML610Q111**
- Short both sets of jumpers to the **RIGHT** for operation of the **ML610Q112**

3. COM Port Selection

When connecting this device to the PC GUI application, please be sure to connect this GUI to the appropriate COM Port of the LAPIS application platform

4. Open Port/Close Port Buttons

These buttons will allow the user to open and close the COM port chosen under section 2. These buttons will open/close the COM Port/UART connection between the GUI application and the LAPIS application platform. **Please note that other tabs (GPIO, I2C, etc.) are not available until the device is connected.**

5. Test UART Button

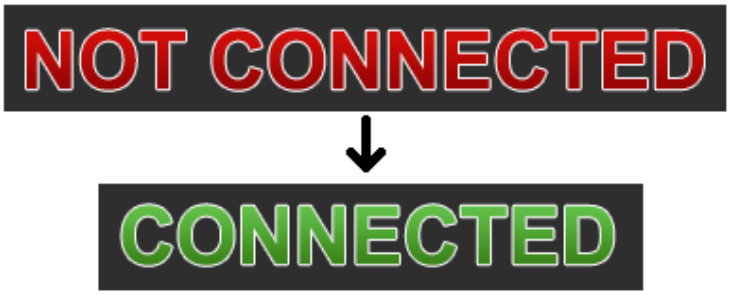
Sometimes, the LAPIS device will be connected through UART; however, the LAPIS's internal programming might not have started/froze/etc. This can be easily fixed by hitting the reset button and clicking the "Test UART" button. Two events should follow for a successful connection:

1. The LAPIS UART Console will return:
   - 

2. Also, the "Connection Indicator" will change from "Not Connected" to "Connected"
   - See below for additional explanation on the connection indicator

6. Connection Indicator

The Connection indicator is used with the "Test UART" Button. When the LAPIS acknowledges the "Ack" message (can occur when "opening the port" or by "Test UART" Button), the indicator will switch from "Not Connected" to "Connected"

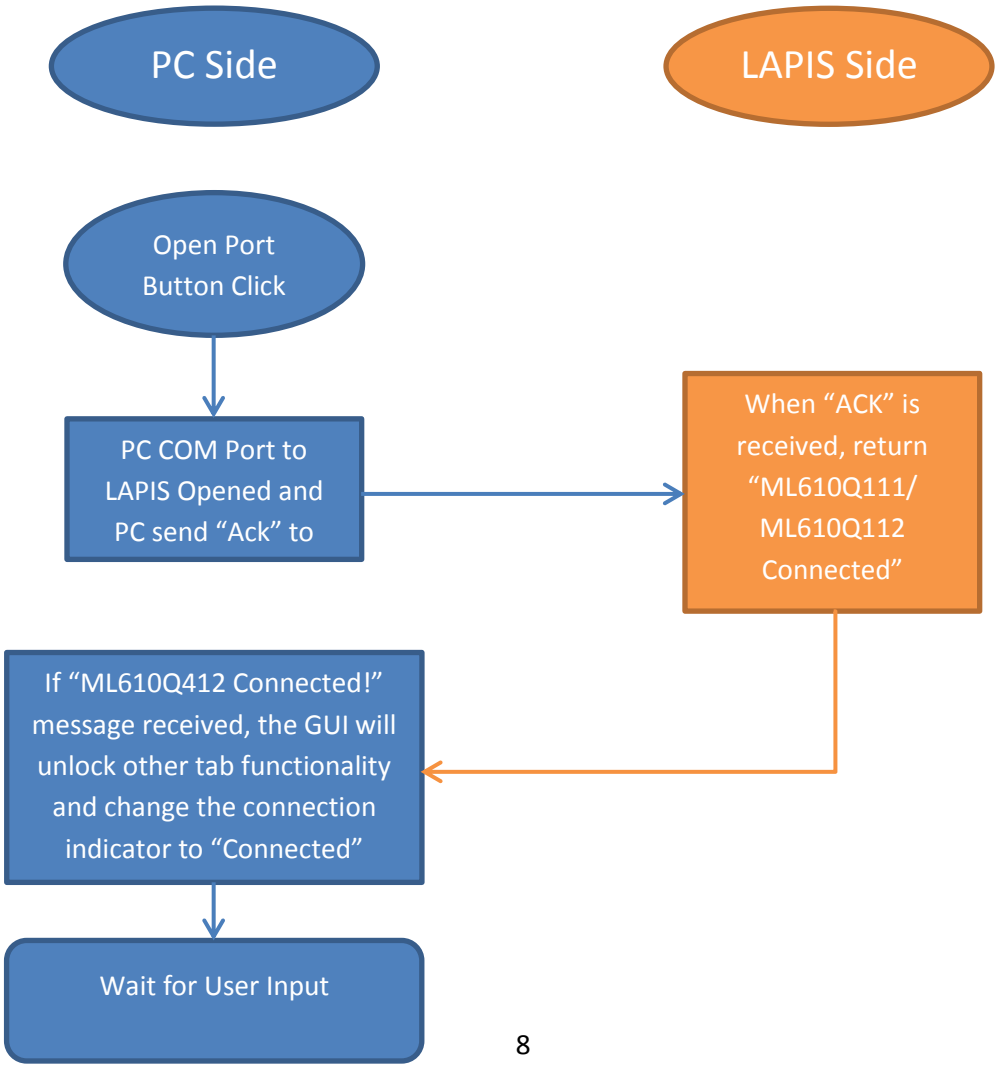


7. Applicable UART Console Information

While in this window, the following UART command sequences should be seen:

- PC to LAPIS Acknowledge
  - PC To LAPIS => Sends **"ACK "** (21 Bytes total) [Spaces as end characters]
    - This command is sent on button presses for *"Open Port"* and *"Test UART"*
  - If LAPIS is connected and sees the above message…
    - LAPIS to PC => Sends **"ML610Q112 Connected!"**

2.1.3 Initial Connection Flow Diagram between GUI and MCU



2.1.4 Connection FAQ:

- I don't see a COM port listed!?
  - Please be sure to check and see that FTDI drivers are installed properly
- COM is connected, but I can't get the "status indicator" to change to "Connected"!?
  - Please reset the MCU and hit the test connect button
  - If failed, please be sure you are connected to the correct COM port

2.2.1 GPIO Tab

Once the user clicks the GPIO Tab, Port C will be shown as default. The GPIO tabs allow the user to adjust the I/O direction for all GPIO pins available for this part number. It allows the user to read all inputs and specify their outputs.

Additionally, please note that these tabs will change according to which device is set in the Config/UART page. The Q111 supports PORT B and C while the Q112 supports Port B, C, and D. Picture references will also change accordingly. However, communication references and operational usage is the same for both MCUs.

Within the tab, there are additional tabs which allow the user to change GPIO pins within the different ports located on the MCU.

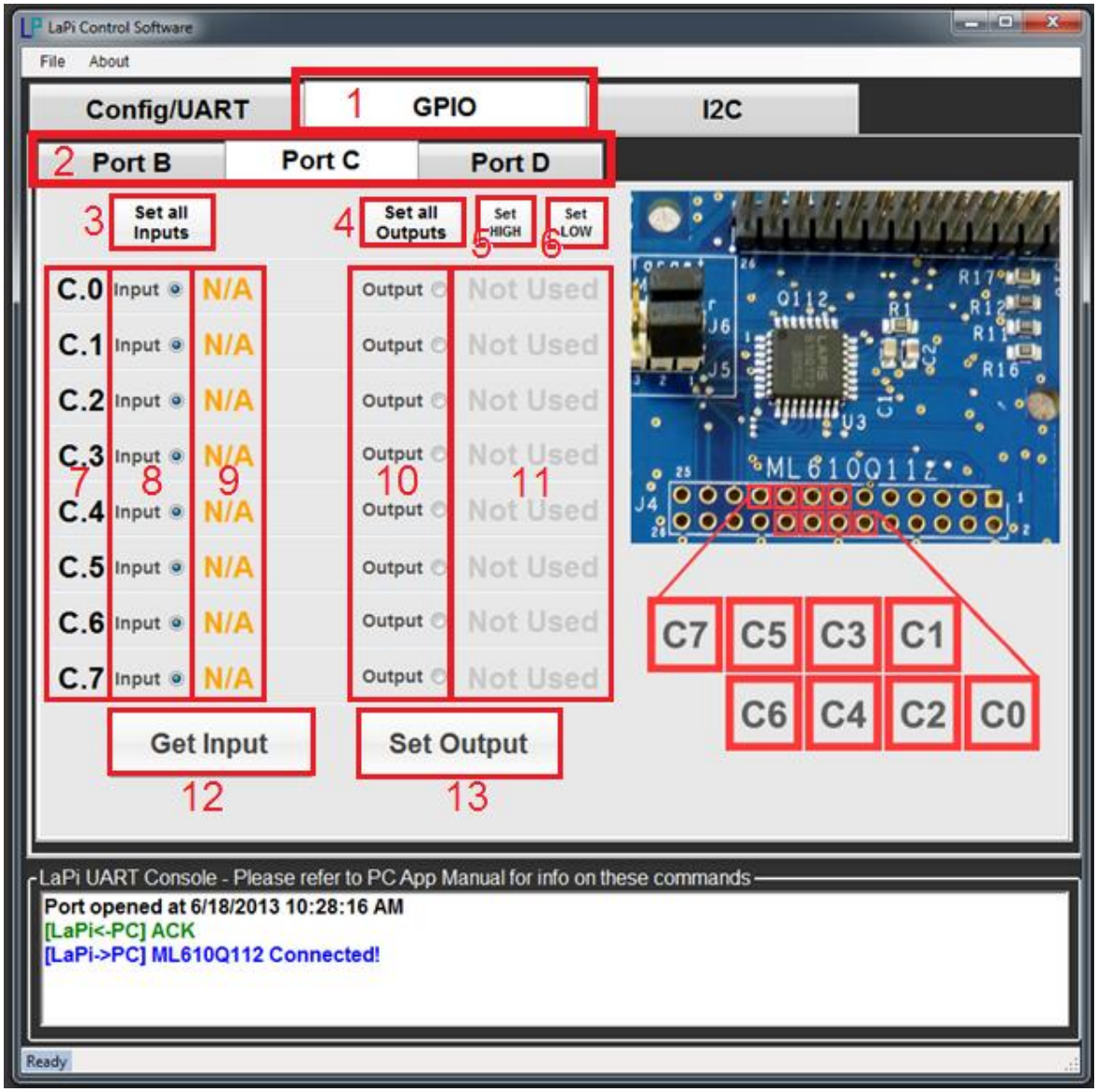


Figure 3: LAPIS GPIO Tab, Port C for ML610Q4112 (Default/Initial View)

2.2.2 *Tab Information*

1. GPIO Tab:

Clicking on the "Config/UART" tab will bring the user back to this screen.

2. GPIO Port Selection Tabs

When using this GPIO tab, the user will be able to control the GPIO pins within Ports B, C, and D. Please use these tabs to switch between the different ports. Please note that the page layouts are similar in function and that the following element descriptions match the different port settings.

3. "Set All Inputs" Button

This Button is used to set all pins within a port as Input GPIO pins. Please note these changes will be updated when the user presses the "Get Input" Button.

4. "Set All Outputs" Button

This Button is used to set all pins within a port as Output GPIO pins. Please note these changes will be updated when the user presses the "Set Output Button"

5. "Set High" Button

This button is used to set all applicable output pins to the "High" level. Please note these changes will be updated when the user presses the "Set Output Button"

6. "Set Low" Button

This button is used to set all applicable output pins to the "Low" level. Please note these changes will be updated when the user presses the "Set Output Button"

7. Port Names

These values indicate the particular GPIO pin that is being modified or controlled

8. Input Pin Radio Button Array

The input pin radio buttons can be used to set pins as input GPIOs. Pressing the "Set All Inputs" button will automatically set all these radio buttons.

9. Input Pin Level Status Field

These labels will be updated upon pressing the "Get Input" button. This field can display the following options:

Input Pin Value	Status Description
High	This means the pin level was read at a "high Level"
Low	This means the pin level was read at a "low level"
Not Used	This means the pin is not set as an input
N/A	This means the pin was set to Input, but the GUI has not read this pin level yet

10. Output Pin Radio Button Array

The input pin radio buttons can be used to set pins as Output GPIOs. Pressing the "Set All Outputs" button will automatically set all these radio buttons.

11. Output Pin Level Settings Field

When the user presses the "Set Output" button, the PC GUI reads the labels set in this field and sends the output pin direction and levels to MCU. Please note that these labels can be clicked to change from High to low, and vice versa. The field can display the following options:

Output Pin Value	Status Description
High	This means the Pin level will be set at a "high Level"
Low	This means the Pin level will be set at a "low Level"
Not Used	This means the Pin is not set as an output

N/A	This means the pin was recently set to Output, but pin level is not yet set on MCU

12. "Get Input" Button

This button is used to send the currently set I/O configuration and request the GPIO input levels of all Input pins set. Pressing this button on the GUI will update the "Input Level Status Fields" based on information received from the UART.

- UART Console information
  - When "Get input" button is pressed, the following example string is sent:
    - Example: "INC_11111111_xxxxxxxx"
    - This string parses into the following:

UART string sent by GUI after "Get Input" Button is pressed						
Example String	IN	C	_	11111111	_	xxxxxxxx
Byte #	[0:1]	[2]	[3]	[4:11]	[12]	[13:21]
Description	Input Requested	Port Indicator	Parsing Character	direction of port, MSB on the left	parsing character	null for input
Options	None	B, C, or D	None	1 -> Input 0 -> Output	None	None

  - When this string is seen by the MCU, the MCU will return two strings
    - String 1 Example: "INC_11111111_01010101
    - This string parses into the following:

UART string sent by MCU after "Get Input" Button String is received						
Example String	IN	C	_	11111111	_	01010101
Byte #	[0:1]	[2]	[3]	[4:11]	[12]	[13:21]
Description	Input Received	ACK Port Indicator	Parsing Character	ACK direction of port, MSB on the left	parsing character	Return Pin Levels of the Port
Options	None	B, C, or D	None	1 -> Input 0 -> Output	None	1 -> High Level 0 -> Low Level

    - String 2: "INC Received"
      - This string follows the previous string and is sent from the MCU to the PC app. This string is a final "acknowledge" message that indicates that the input strings were sent properly.

13. "Set Output" Button

This button is used to send the currently set I/O configuration and also set the GPIO input levels of all output pins set. Pressing this button will take Pin level values from the "Output Pin Level Settings Fields" and set them on the MCU using the UART connection.

- UART Console information
  - When the "Set Output" button is pressed the following example string is sent:
    - "OUC_00000000_00000000"
  - This string parses into the following:

UART string sent by GUI after "Set Output" Button is pressed						
Example String	OU	C	_	00000000	_	00000000
Byte #	[0:1]	[2]	[3]	[4:11]	[12]	[13:21]
Description	Output Requested	Port Indicator	Parsing Character	Direction of port, MSB on the left	Parsing Character	Indicates Pin Level of the Port
Options	None	B, C, or D	None	1 -> Input 0 -> Output	None	1 -> High Level 0 -> Low Level

  - When this string is seen by the MCU; the MCU will return one string
    - String 1 Example: "OUC Received 01010101"
    - As the PC GUI is setting the output, the returned string only acknowledges that the Output command was sent. Thus, the return string parses into the following:

UART string sent by MCU after "Set Output" Button String is received				
Example String	OU	C	Received	01010101
Byte #	[0:1]	[2]	[3]	[13:21]
Description	Input Received	ACK Port Indicator	ACK Phrase	ACK Pin Levels of the Port
Options	None	B, C, or D	None	1 -> High Level 0 -> Low Level

2.2.3 Other GPIO Tab Information

Please note that the tab configuration will change depending on the device selected in the Config/UART tab. The ML610Q111 supports Ports B and C while the ML610Q112 supports ports B, C, and D. Please note that even though Ports A and D contains similar structures as the previously described Port C, there are differences in the number of pins that can be accessed.

Port B is depreciated due to several pins used for other functions (UART, I2C, SPI). This tab is accessible by both ML610Q4111 and ML610Q4112.

Port D only contains 6 usable GPIO pins for this particular chip. This tab is only accessible by the ML610Q4112.

Please see the figures below for pictures of the other GPIO port tabs.

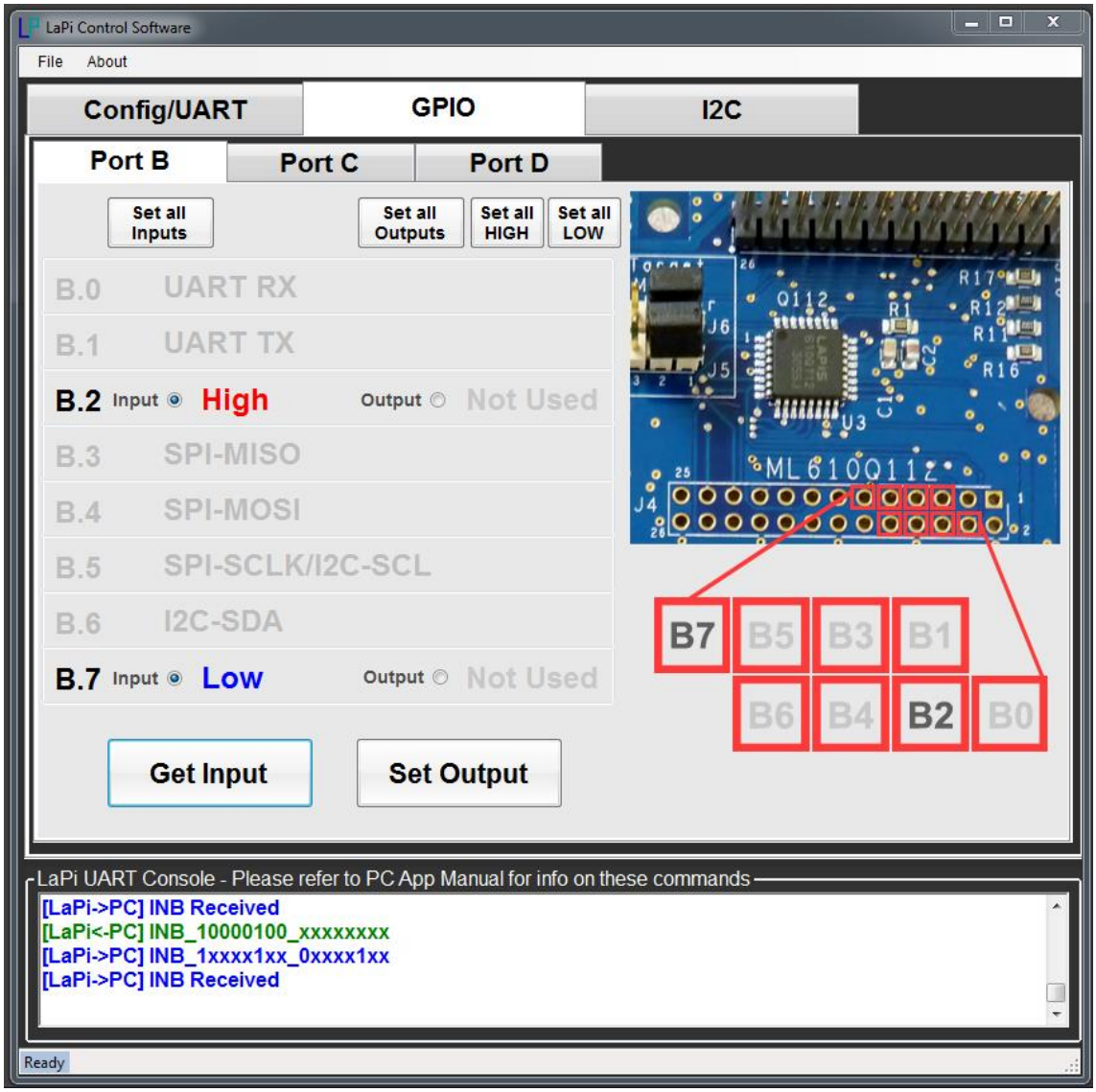


Figure 4: LAPIS GPIO Tab; ML610Q112, Port B

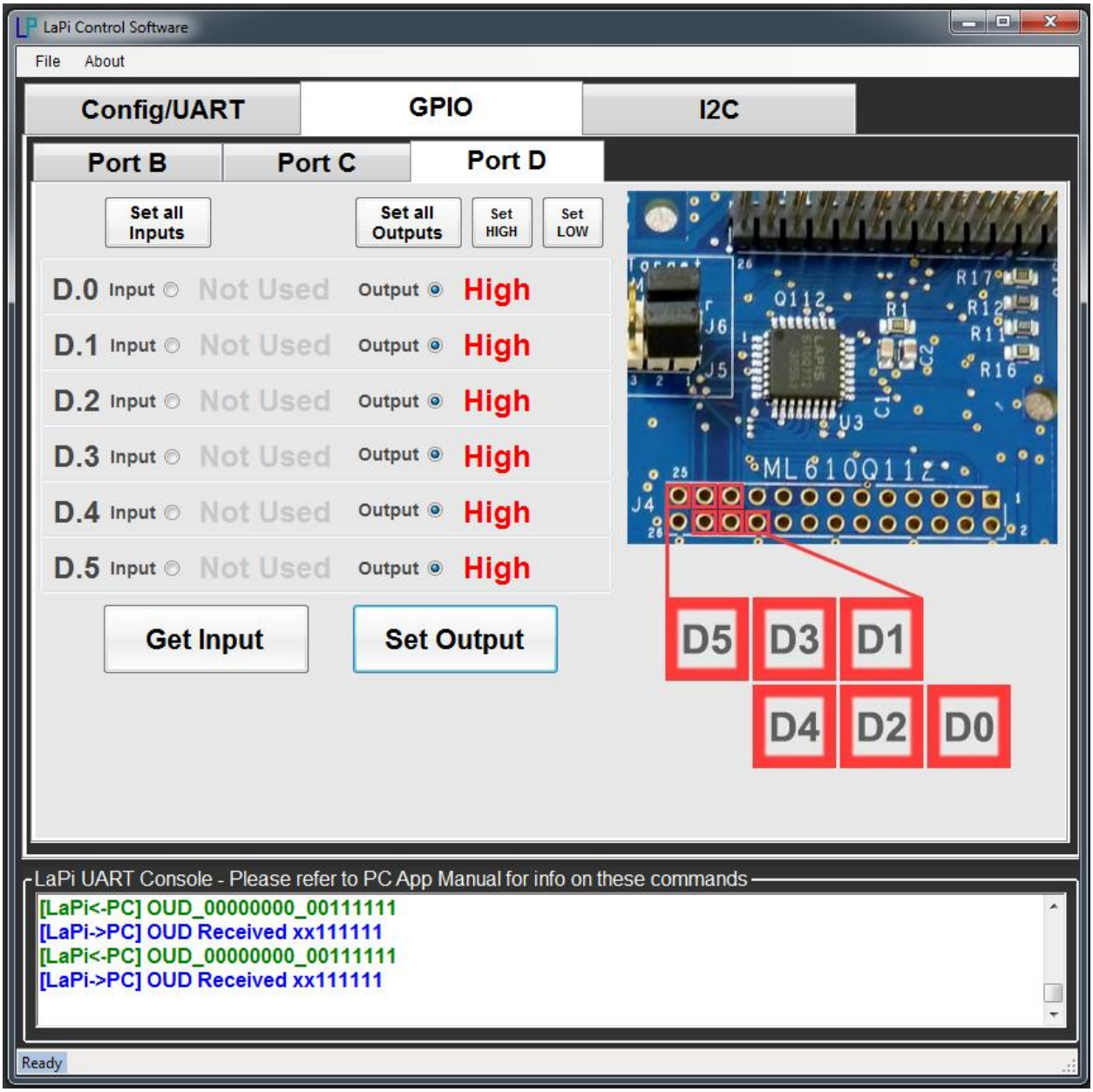


Figure 5: LAPIS GPIO Tab; ML610Q112, Port D

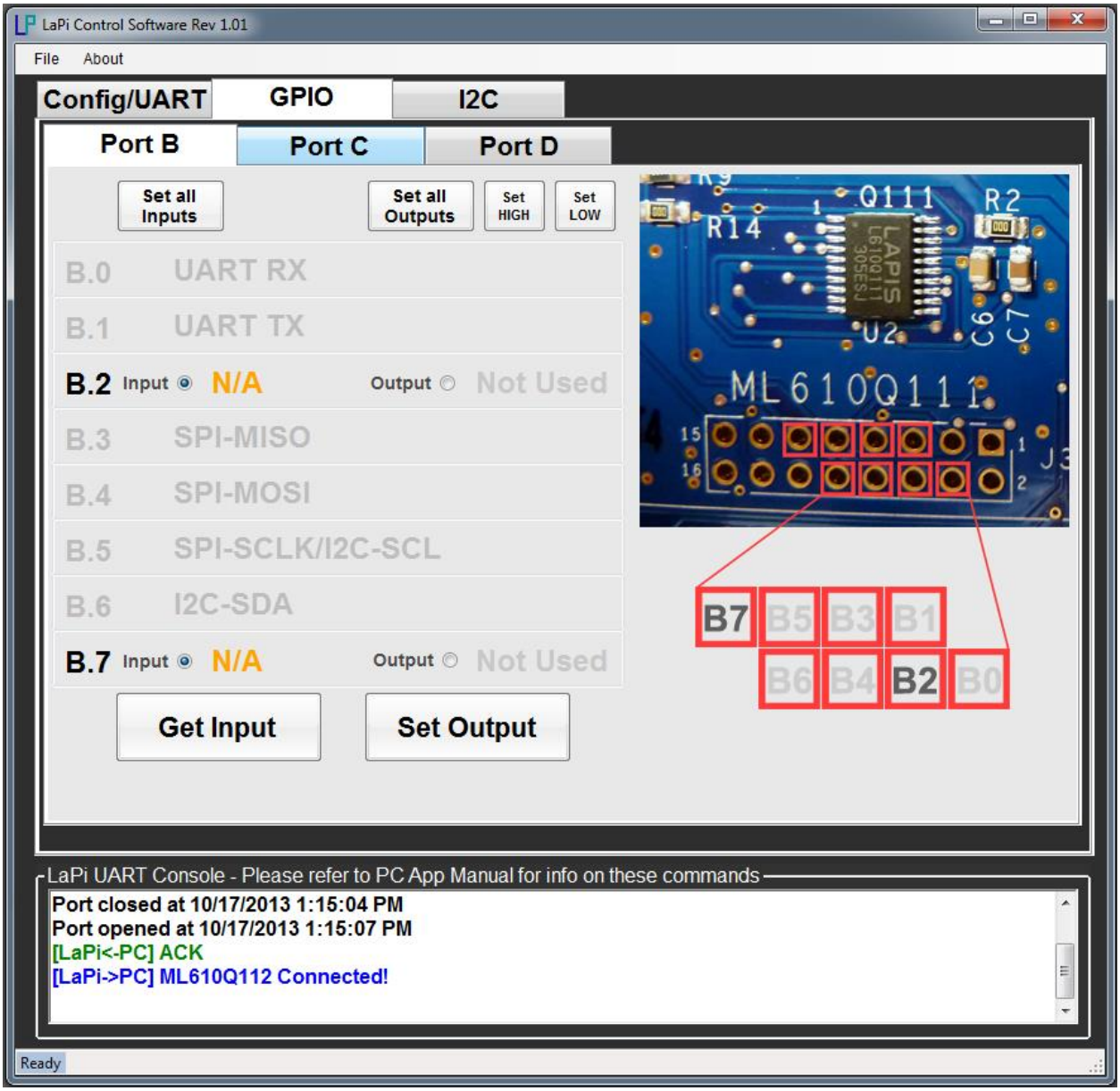


Figure 6: LAPIS GPIO Tab; ML610Q111, Port B

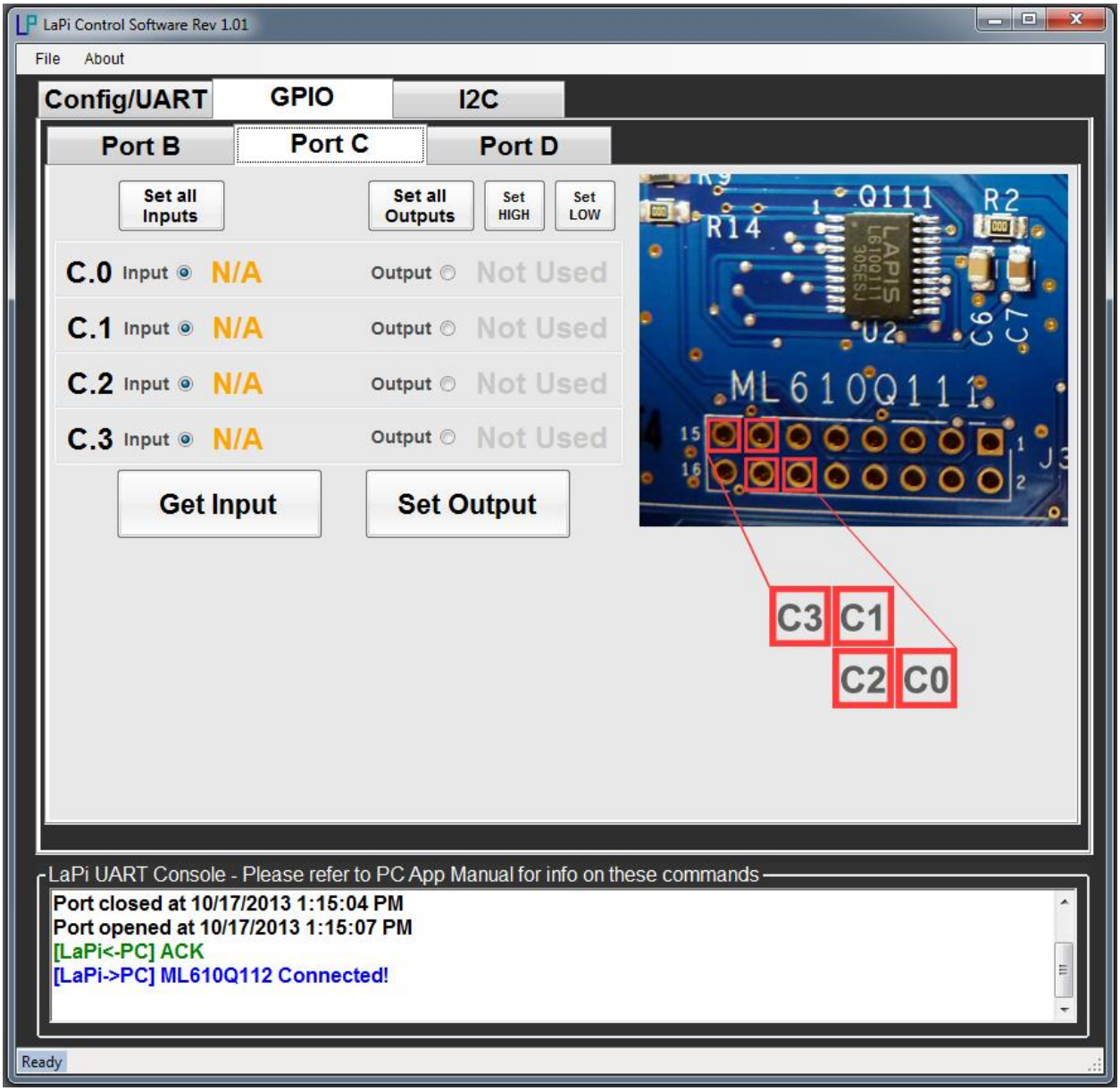


Figure 7: LAPIS GPIO Tab; ML610Q111, Port C

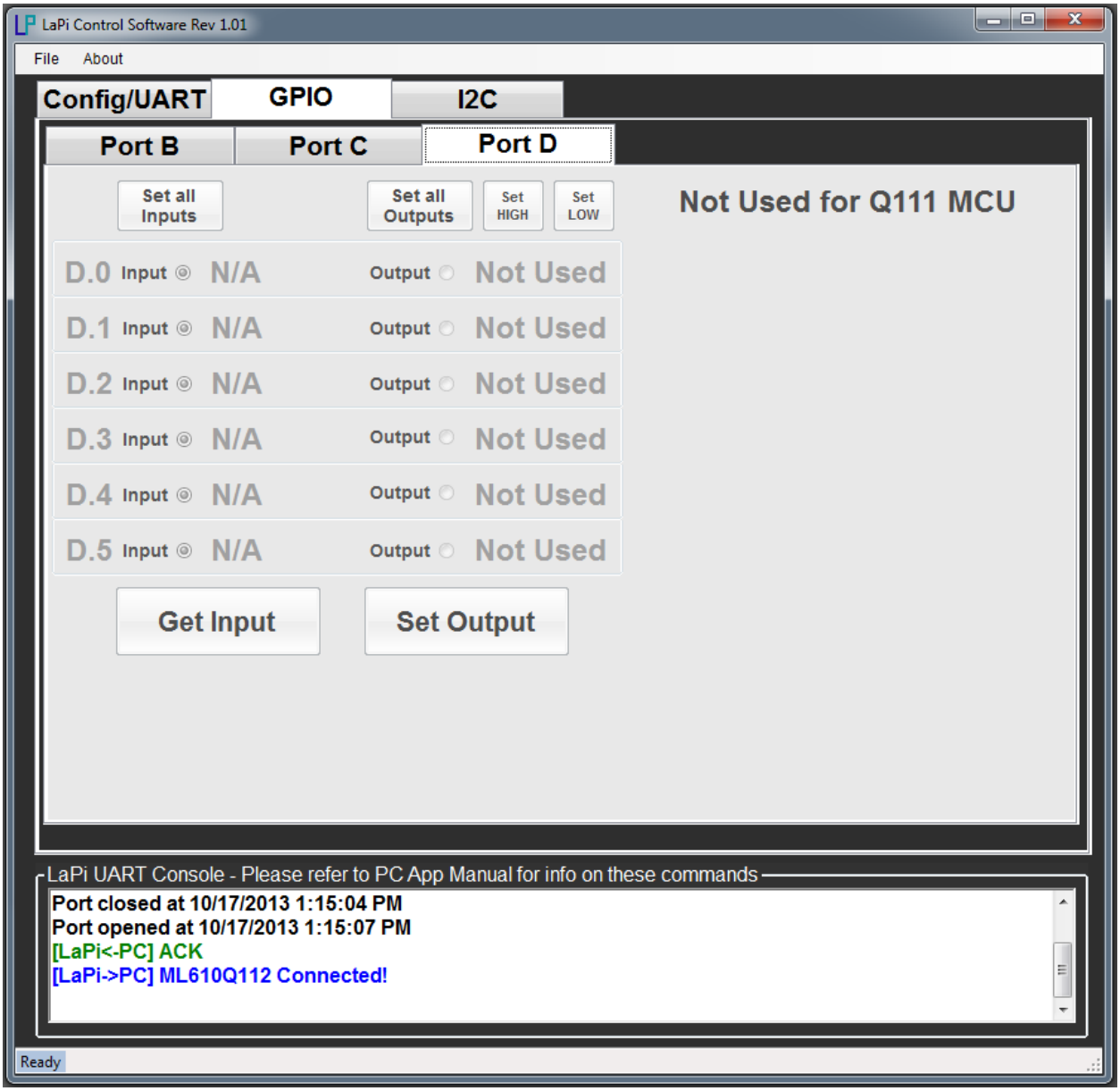


Figure 8: LAPIS GPIO Tab; ML610Q111, Port D (Not Accessible)

2.2.1 I2C Tab

Once the user clicks the I2C Tab, the tab shown in figure 9 below will be displayed. This tab will allow the user to read and write to the I2C bus on the target microcontroller.

Additionally, please note that these tabs will change according to which device is set in the Config/UART page. For I2C, only the picture references will change accordingly. All communication references and operational usage is the same for both MCUs.

Please note that there are no internal pull up resistors for the I2C rails, so when connecting this device, please be sure to add the appropriate pull up resistance for your application.

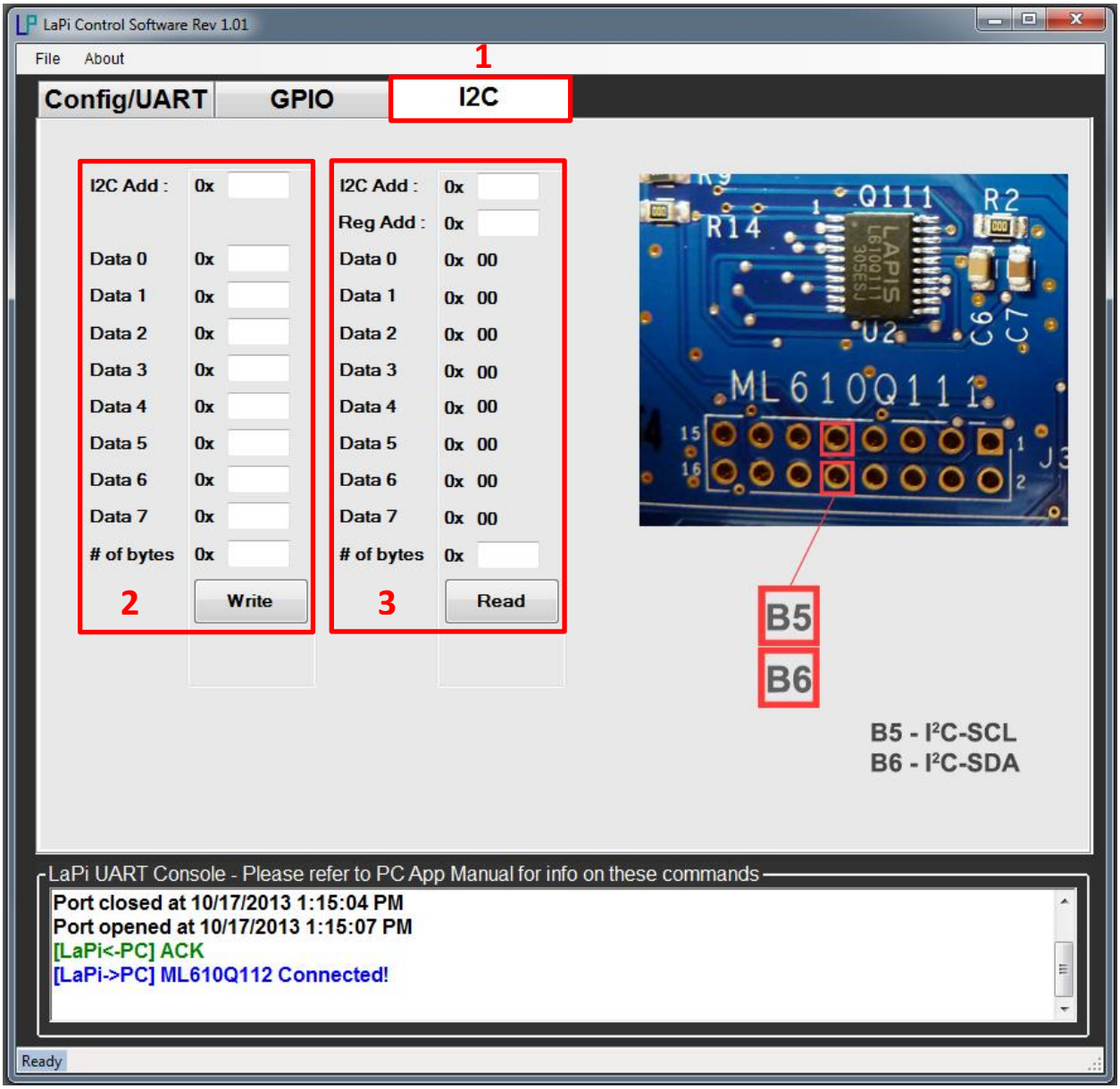


Figure 9: LAPIS I2C Tab for ML610Q4111 (Default/Initial View)

2.2.2 Tab Information

1. I2C Tab:

Clicking on the "I2C" tab will bring the user back to this screen.

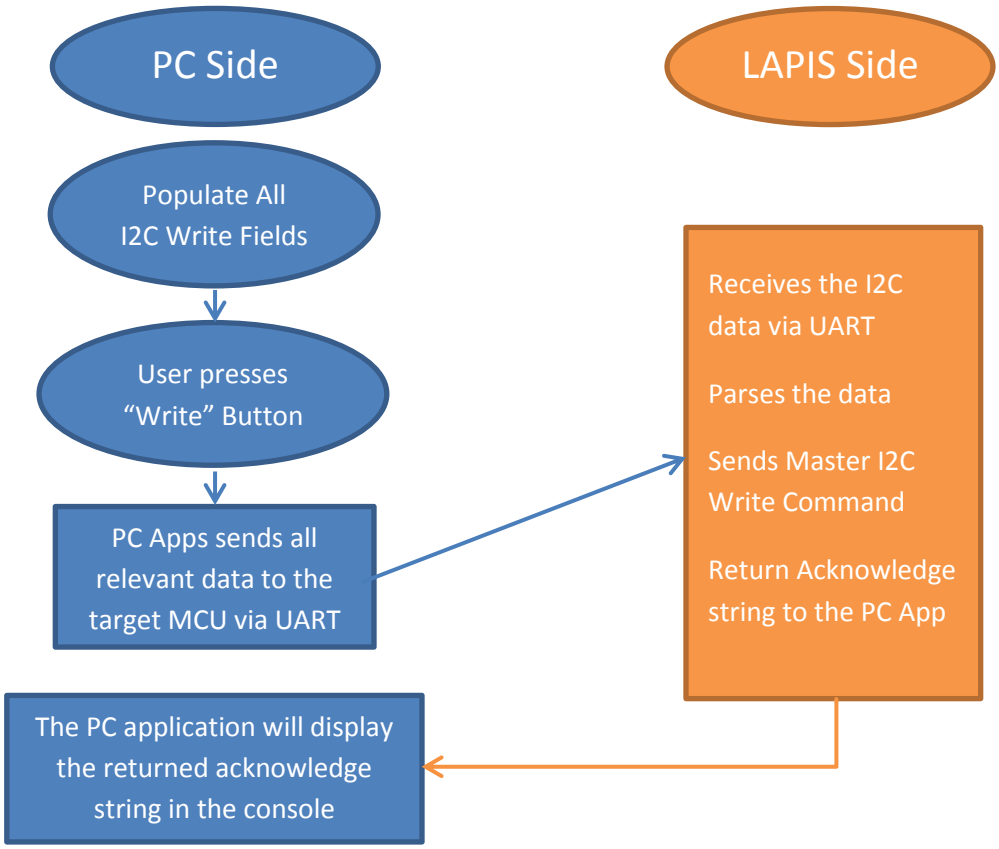
2. I2C Write

This section allows the user to send I2C Write commands through the PC Application.

- User Input data fields
  - "I2C Add" Field
    - This field should be populated with the 7bit device address you would like this device to talk to
  - "Data0 to Data7" Fields
    - These Fields should be populated with the data you want sent directly after the I2C device address is sent.
  - "# of bytes" Field
    - This field should contain the number of data bytes you want to send to the device.

For the I2C Write, the communication for sending the I2C commands follows the below flow diagram on the next page:

Master I2C Write Flow Diagram



3. I2C Read

This section allows the user to send I2C Read commands through the PC Application

- User Input data fields
  - "I2C Add" Field
    - This field should be populated with the 7bit device address you would like this device to talk to
  - "Reg Add" Field
    - These fields should be populated with the device's register address you would like to read from. For default usage, please enter 0x00 here.
  - "# of bytes" Field
    - This field should contain the number of data bytes you want to read from the device.

For the I2C Read, the communication for sending the I2C commands follows the below flow diagram:

Master I2C Write Flow Diagram