

**LAPIS
Mini Low
Power MCU
Development
Board**

USER MANUAL

CONTENTS

INTRODUCTION	5
LAPIS Development Board Objective	5
QUICK START GUIDE	6
BOARD OVERVIEW	7
LAPIS MICROCONTROLLER OVERVIEW	8
ML610Q111	8
ML610Q112	8
LAPIS FUNCTIONALITY	9
Mini Low Power Microcontroller Features	9
UNIVERSAL SOCKETS & PLUG-IN BOARDS	10
Prototyping Space	10
MICROCONTROLLER SELECTION JUMPER SETTINGS	11
Overview of Ports, Sockets & Selection Jumpers	11
J2 - Voltage Level Selector	11
(3.3V or 5.0V)	11
J14 & J7 - Debug Control	12
J5 & J6 - Target Micro Selector	12
J12 & J13 - UART COMM Selector	13
nanoEASE Debugger Connection	13
MICROCONTROLLER GPIO & UNIVERSAL SOCKETS	14
Prototyping Area	15
nanoEASE MICROCONTROLLER PROGRAMMER & DEBUGGER	15
MICROCONTROLLER RESET BUTTON	16
USB TO SERIAL UART INTERFACE	16
USB Driver Support	16

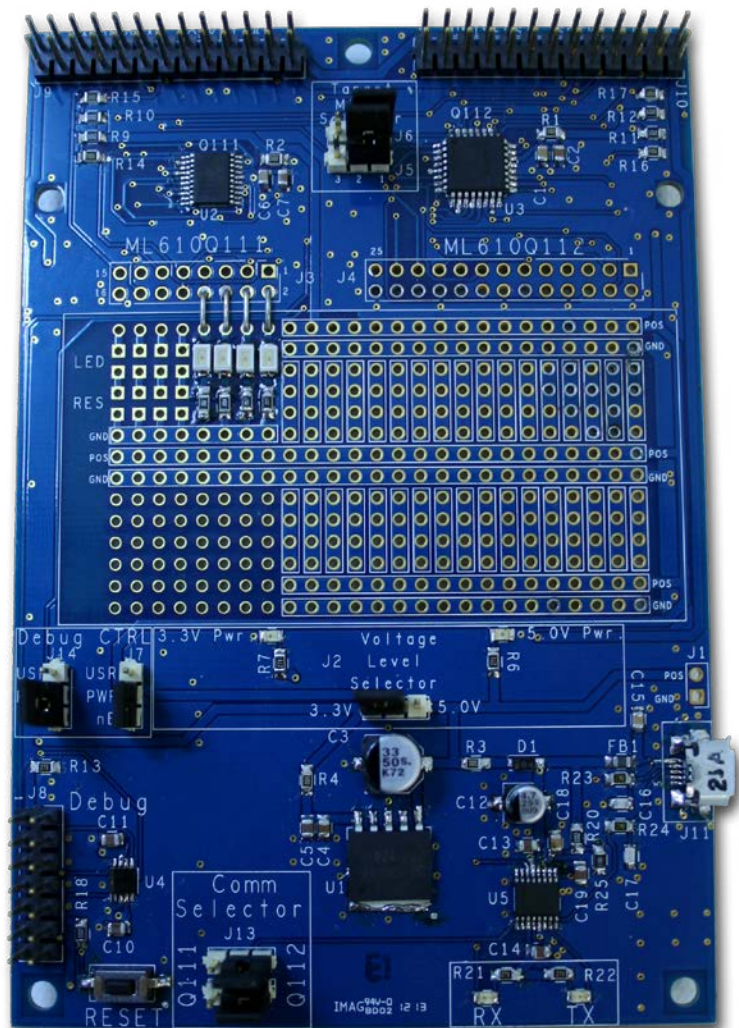
LAPIS GRAPHICAL USER INTERFACE (GUI)	17
Configuring the Interface	17
Selecting the GPIO Tab	18
Controlling GPIO Outputs via GUI	19
Controlling Outputs	19
Reading Inputs	19
LAPIS Q111 PINOUT GUIDE	20
Pin Layout of the ML610Q111 TSSOP20 Package	20
J3 - 15 I/O Pins of the Q111 Micro	20
LAPIS Development Board Pinout Details for the Q111 Micro - cont'd	21
J9 - Q111 Universal Socket #1 w/Pinout (Compatible w/Raspberry Pi® GPIO Layout)	21
Raspberry Pi	21
GPIO Port	21
Reference Information	21
LAPIS Q112 PINOUT GUIDE	22
Pin Layout of the ML610Q112 LQFP32 Package	22
J4 - 25 I/O Pins of the Q112 Micro	23
J10 - Q112 Universal Socket #2 w/Pinout (Compatible w/Raspberry Pi® GPIO Layout)	23
Q111 & Q112 PWM CONFIGURATION OPTIONS	24
APPLICATION EXAMPLE FOR Q111 & Q112 MICROS	25
LAPIS DEVELOPMENT BOARD SCHEMATICS	26
Input power Block	26
nanoEASE Block	27
Q111 Micro Block	28
Q112 Micro Block	29
Universal Sockets Block	30
USB to UART Circuit Block	31
LAPIS DEVELOPMENT BOARD BILL OF MATERIAL (B.O.M)	32
(BOM Page 1 of 2)	32
(BOM Page 2 of 2)	33

INTRODUCTION

LAPIS Development Board Objective

The LAPIS development board is designed to make it easy to create circuits and programs around LAPIS' powerful Low Power Microcontrollers.

This guide will help you understand the basic features of our microcontrollers, show you how to set up your new LAPIS Development Board, and help you get started creating circuits built around this platform. You will find it easy to begin creating custom add-on boards. You can also check out add-on boards we'll be making available in the future to better demonstrate the power, flexibility and possibilities of these great micros.



QUICK START GUIDE

1. Your LAPIS board ships with jumpers pre-configured so that the 112 communicates with the USB interface, the power is provided by USB and the RESET is controlled by the button on the board (see Section 7 for further details). Verify that the jumpers should be set so that:

- a. The Q112 micro will communicate with the LAPIS GUI (J12 & J13 = 112)
- b. POWER provided by USB (J2 controls 5V or 3.3V, J7 = USR USB power)
- c. RESET controlled by the nanoEASE Debugger or Reset button (J14)

Steps 2 through 5 explain how to use the PC based GUI to control the ML610Q112

2. The GUI along with a program on the device that understands the USB GUI's instructions can be used to control the basic functionality of the device without the use of the debugger. If you are going to use the GUI, install FTDI Virtual Com Port USB drivers (see Section 11 for further details) from the FTDI driver support website:

<http://www.ftdichip.com/Drivers/VCP.htm>.

- a. The Reset jumper (J14) should be in the USR position.
3. Copy the contents and subdirectories of the directory GUI from the USB stick, which is included with this kit, to a GUI directory on your computer. The sub folders contain the source code and detailed documentation for the GUI application.
- a. The documents provided with this kit may not be the most up to date. To check go to the LP Micro landing page (<http://www.rohm.com/web/global/low-power-microcontroller>) to see if there is a more recent code release
4. Connect the LAPIS board to a computer via the provided micro USB cable and run the LAPIS GUI. See Section 12 or the LAPIS GUI User's Manual for details.

5. Start the GUI by clicking on the program **LAPIS_ControlSoftware_6-28-2013_Rev1p01.exe** which is in the GUI folder that you copied from the USB stick to your computer.

Steps 6 through 9 explain how to use the SW Design, Development and Debugging tools to control the basic operation of the ML610Q112. The debugger can be used with the PC based GUI application or any other application code running on either of the Mini LP Micros on the board.

6. To control and modify the code that is running on the ML610Q112, install the U8 Code Development tools from the CD included with the nanoEASE. (the U8Dev Suite includes 18 manuals to help you become familiar with LAPIS' micros, the development tools, the nanoEASE debugger, and more). Follow the instructions in the ReleaseNote_e.pdf while installing the tools.

a. The tools provided with this kit may not be the most up to date. To get the latest LAPIS U8 Dev. Suite go to this website and register and download the latest version of the tools: <https://www.lapis-semi.com/customer/lpmcu/login.html>

7. The source code, a detailed description of the functionality of the code for the program that is already on the device, other sample projects, and other documents such as PCB design files, daughter board templates are provided on the USB stick that is included with this kit. Copy the contents of the Projects directory from the USB stick to a Projects directory on your computer.

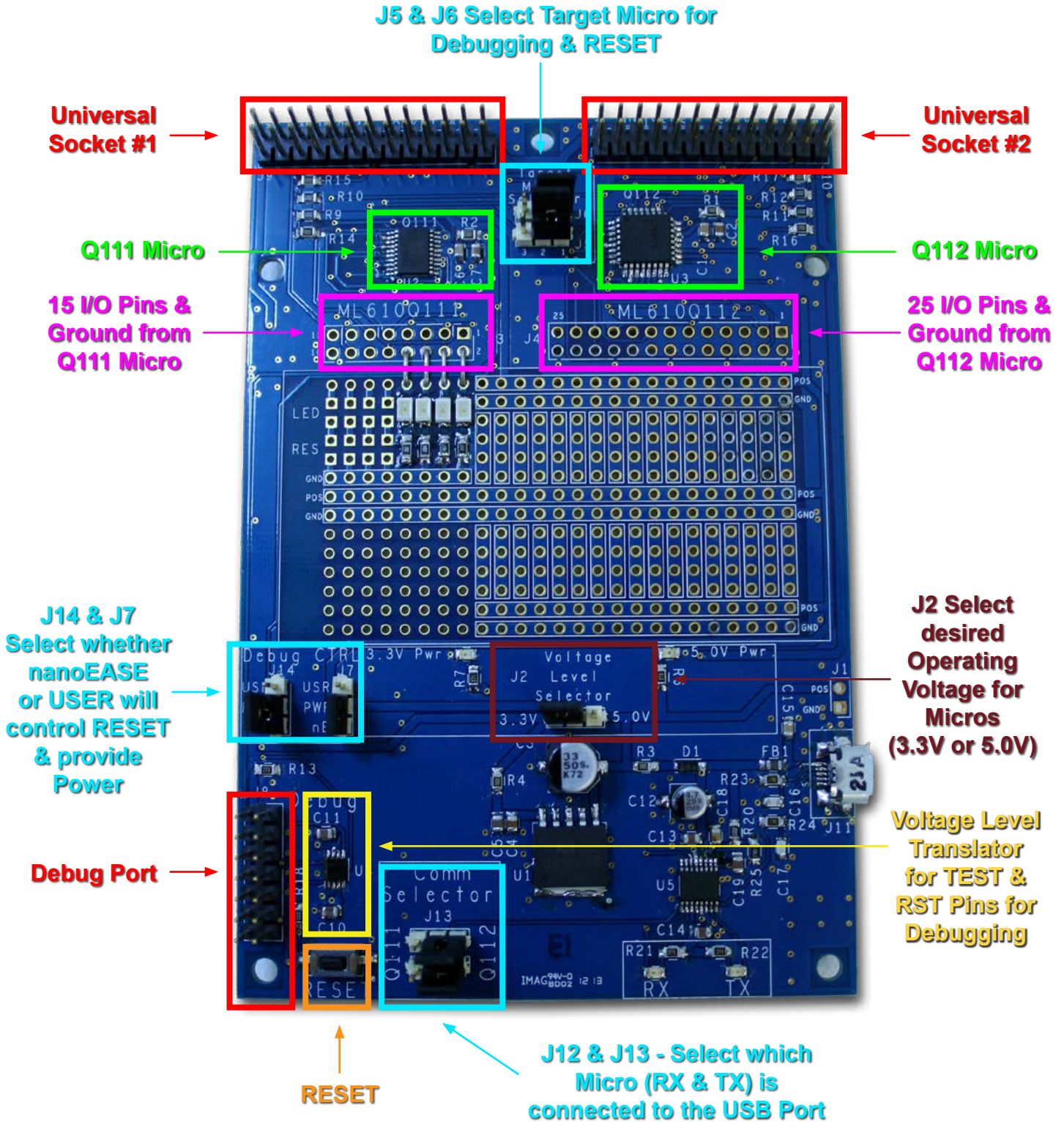
a. The documents provided with this kit may not be the most up to date. To check go to the LP Micro landing page (<http://www.rohm.com/web/global/low-power-microcontroller>) to see if there is a more recent code release.

8. Using the U8 IDE icon placed on your Desktop or **Start -> Program Files -> U8 Tools -> nX-U8 -> IDEU8 GUI Environment** to start the IDEU8 Code Development Environment for the LAPIS micros.

a. Using File -> Open the project file Q112-GPIO.PID in the directory ...\\Projects\\...
xxx\\

9. Connect the nanoEASE to the board, being careful to align it correctly per the instructions in the section of this document titled nanoEASE Debugger Connection.
10. Start the DTU8 Debugger by clicking on **Project » Debug**. Refer to the Debugger manual for detailed instructions on using the debugger.
11. Send an e-mail to the LAPIS Development Kits support team LAPIS_Dev_Kit_Support@rohms Semiconductor.com for answers to questions that you may have about this kit or the associated devices.

BOARD OVERVIEW



LAPIS MICROCONTROLLER OVERVIEW

There are two built-in microcontrollers on this development board:

1. **ML610Q111** (referred to in this document as the Q111) – a 24kB micro in a 20-pin plastic TSSOP package
2. **ML610Q112** (referred to in this document as the Q112) – a 32kB micro in a 32-pin LQFP package

The Q111 features 15 I/O pins and the Q112 has 25 I/O pins.

LAPIS low power microcontrollers are high-performance 8-bit CMOS microcontrollers with a rich assortment of built-in peripheral circuitry and internal low speed and high speed oscillators that allow for operation up to 8MHz.

They contain both 8-bit & 16-bit timers, 4-channel 16-bit PWM (including an option for 2, 3 or 4 synchronous outputs), serial interfaces such as UART, SSIO (Master or Slave mode; can also be used for SPI), and I²C (Master or Slave Mode), two-channel analog comparators, voltage level supervisor (VLS) functionality, 10-bit successive approximation type A/D converters, and an interrupt controller with 31 interrupts including a watch dog timer – all incorporated around an 8bit nX-U8/100 CPU.

The nX-U8/100 CPU is capable of very efficient instruction execution (one instruction per clock cycle via pipeline architecture using parallel processing). The on-chip debug function enables programming as well as program debugging.

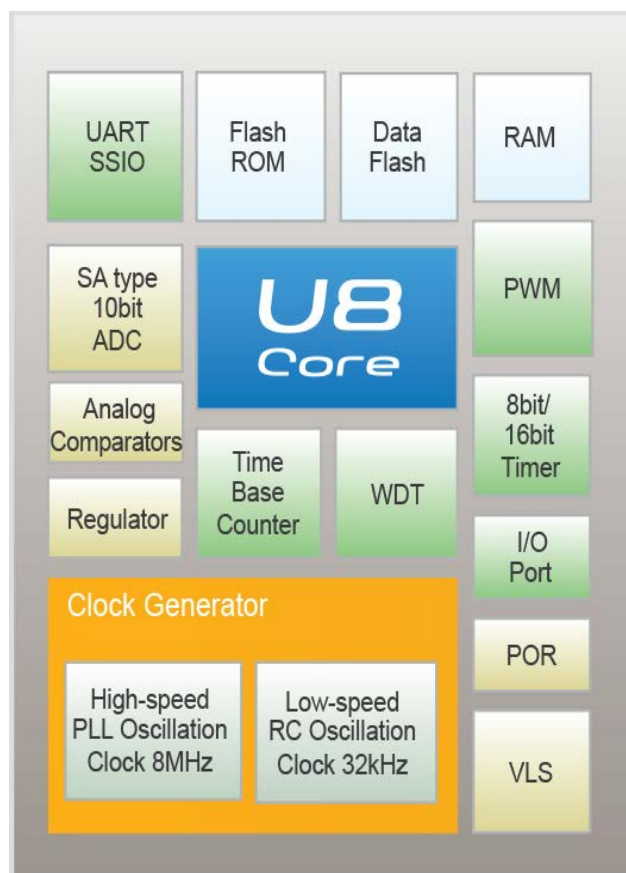
Both micros have a very broad operating voltage range from 2.7V to 5.5V, allowing for easy operation at convenient standard voltages of either 3.3V or 5.0V. They also have a guaranteed operating temperature range of -40°C to 125°C (-40°F to 257°F).

LAPIS FUNCTIONALITY

The board is designed so that by selecting the appropriate jumper settings either microcontroller can interface to the low-cost nanoEASE programmer/debugger pod. Using the free Windows-based CCU8 Compiler, DTU8 Debugger, and FLASH programming utilities the device can be programmed and code can be written, compiled, and debugged in real-time through a 2-wire interface. All I/O pins are routed to a prototyping area on the development board, and there are two universal sockets with common pin functions for interfacing with planned future ROHM development boards that will contain sensors, LED drivers, EEPROM's, and other components. There is also a built-in USB to UART interface that allows easy communication between any computer and the microcontrollers over USB.

Mini Low Power Microcontroller Features

The two micros included in this evaluation board are extremely feature-rich. The clock generator circuit generates and provides a low-speed clock (LSCLK), high-speed clock (HSCLK), PLL clock (PLLCLK), high-speed oscillation clock (OSCLK), system clock (SYSCLK), low-speed output clock (LSCLK), and high-speed output clock (OUTCLK). The integrated high-accuracy RC oscillator features built-in capacitors, along with a voltage regulator that does NOT require external capacitors, contributing to greater miniaturization and lower costs.



Key Features

Mini Low Power Microcontrollers

- 8-bit & 16-bit Timers
- 4-channel 16-bit PWM (including an option for 2, 3 or 4 synchronous outputs)
- Serial I/F
 - UART (2 ch.)
 - SSIO (8/16 Bit mode; Master or Slave)
 - I²C (100kHz or 400kHz; Master or Slave)
- Two-channel analog comparators
- 31 interrupts
- Voltage Level Supervisor (VLS) functionality
- 10-bit successive approximation type A/D converters
- Watch Dog Timer
- Standby current consumption: 1μA in STOP mode

CPU

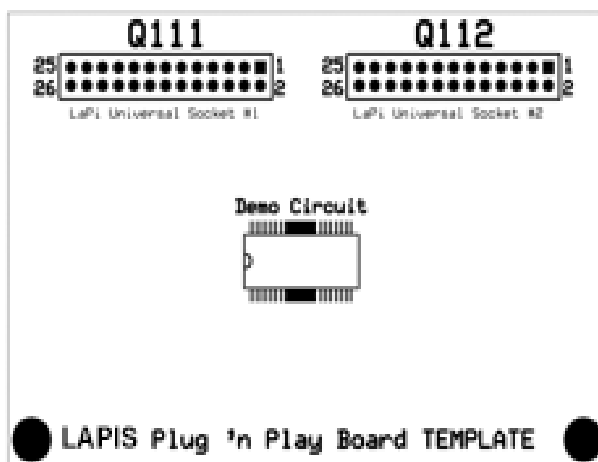
- 8-bit RISC CPU (nX-U8/100)
- 16-bit instruction system
- One instruction per clock cycle using parallel processing via pipeline architecture
- Minimum instruction execution times:
 - 30.5μs (internal RC oscillator@32.768kHz)
 - 0.122μs (internal PLL@8.192MHz)

LAPIS FUNCTIONALITY

The development board features two universal sockets that are compatible with the Raspberry Pi® expansion ports (one for each micro) with common pin functions for easy interfacing with daughter boards. Daughter boards can be easily designed to mate with these universal sockets and plugged into the evaluation board for validation of your design. If a more affordable, lower power processor is required than is provided on a Raspberry Pi® base board, an existing Raspberry Pi® expansion boards can be used with the LAPIS Development kit. The interface contains all of the signals typically necessary to communicate with a wide variety of ICs and sensors.

Plug 'n Play Board Ideas...

- LED Driver Board
- EEPROM Board
- Clock/Timer IC Board
- PMIC Board
- Beam Forming IC Board
- MP3 Decoder Board
- PIR Sensor Board
- Hall Sensor Board
- Cap. Switch Sensor Board
- ALS Sensor Board
- I/O Expander Board
- Transceiver Board
- Image Connection Board
- InDA Board
- Relay Driver Board
- ...and so on



STANDARDIZED PINOUT	
01 - 3.3V	
02 - 5.0V	
06 - GND	
08 - UART RX	07 - GPIO 04
10 - UART TX	11 - GPIO 17
03 - I2C SDA	13 - GPIO 21
05 - I2C SCL	15 - GPIO 22
	12 - GPIO 18
23 - SPI SCLK	16 - GPIO 23
19 - SPI MOSI	18 - GPIO 24
21 - SPI MISO	22 - GPIO 25
24 - SPI CS0	
26 - SPI CS1	
09 - nc	04 - nc
17 - nc	14 - nc
25 - nc	20 - nc

Prototyping Space

In addition to the universal sockets, the evaluation board has all I/O pins (15 I/O pins of the Q111 and 25 I/O pins on the Q112) routed to the edges of this prototyping area, which is designed to allow for easy and rapid custom prototyping with convenient access to all combined 40 I/O pins as well as power and ground rails.

These LAPIS micros can tolerate a voltage input ranging from 2.7V to 5.5V. The evaluation board is set up to allow the micros to be powered from either 5.0V or from 3.3V. There is a jumper selector (J2) for setting the desired voltage level (regulator input voltage – typically 5.0V OR regulator output voltage of 3.3V). The source of the 5.0V would typically come from the USB from a computer, and should thus be limited to no more than 500mA to prevent pulling too much current from your computer port. The 5.0V from the USB bus is fed into a ROHM (BA33D18HFP) 3.3V low drop-out voltage regulator to provide the 3.3V microcontroller powered option. Finally, there is also an option to input any voltage from 4.1V up to 16V into the input of the 3.3V regulator (at J1). To do this, you should first remove R3 (0Ω resistor) to fully isolate VBUS from the input of the LDO voltage regulator.

MICROCONTROLLER SELECTION JUMPER SETTINGS

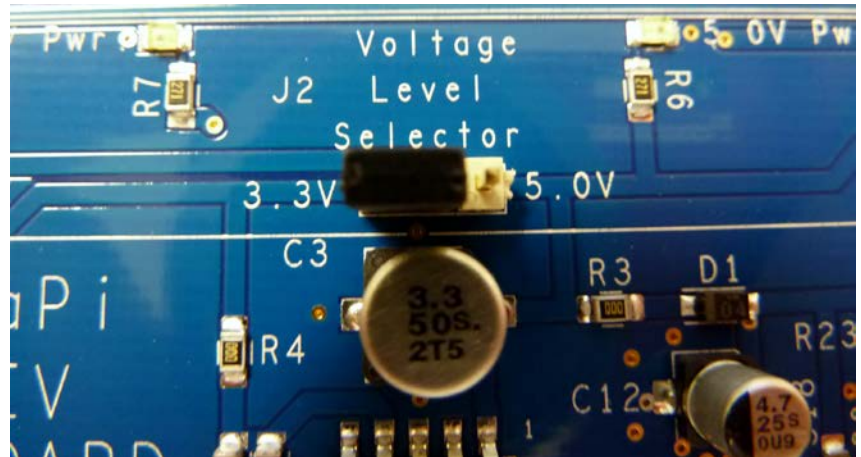
There are 4 jumper selectors (J5, J6, J12 & J13) for setting which micro you are interfacing with. Jumpers J5 & J6 set the interface between the nanoEASE Debugger and the target micro (TST & RST pins), while jumpers J12 & J13 select which micro the USB to UART IC interfaces with (RX & TX pins).

Overview of Ports, Sockets & Selection Jumpers

- **J1** External Power Option
- **J2** Voltage Level Selector (3.3V or 5.0V)
- **J3** All 15 Q111 I/O Pins (plus Ground)
- **J4** All 25 Q112 I/O Pins (plus Ground)
- **J5** Target Micro (TEST Pin)
- **J6** Target Micro (RESET Pin)
- **J7** POWER Select (nanoEASE or User Selected)
- **J8** nanoEASE Debugger Header
- **J9** Universal Socket #1 (Q111)
- **J10** Universal Socket #2 (Q112)
- **J11** MicroUSB Socket
- **J12** COMM PORT RX Selector (USB to Q111 or Q112)
- **J13** COMM PORT TX Selector (USB to Q111 or Q112)
- **J14** RESET Select (nanoEASE or RESET Button)

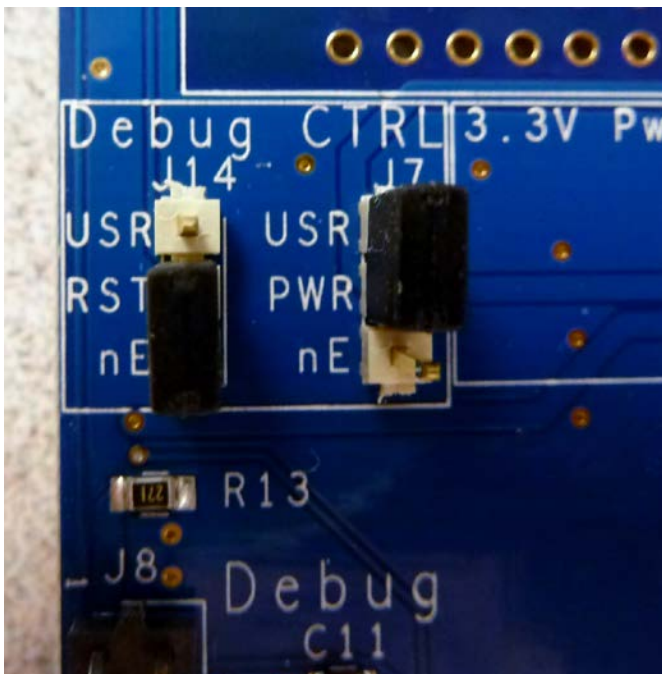
J2 - Voltage Level Selector: (3.3V or 5.0V)

As mentioned above these microcontrollers can be powered from either 3.3V or 5.0V. This jumper selects the microcontroller power input voltage between the input to the voltage regulator (typically 5.0V from the USB port or whatever external input voltage you provide on J1) and the 3.3V output.



In this example, 3.3V from the output of the voltage regulator (500mA) is selected to power the microcontrollers.

J14 & J7 - Debug Control

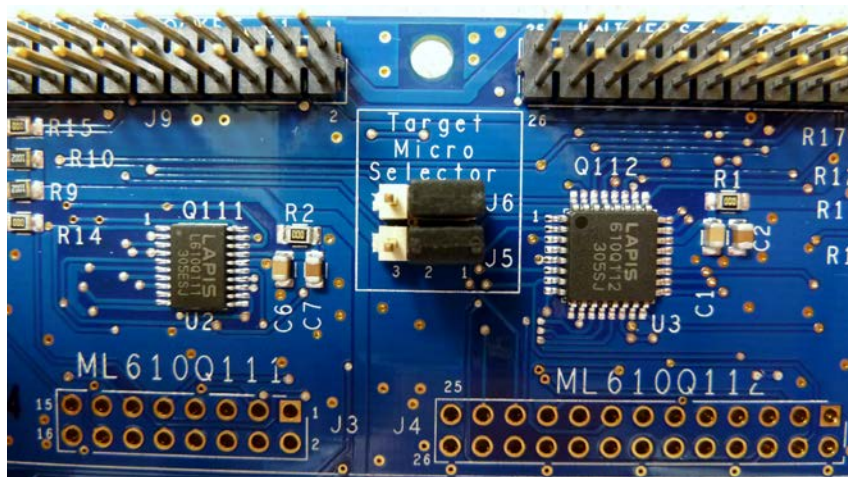


The debugger functionality of these microcontrollers is very useful. You will be able to single-step through your code to quickly find problems and solutions. You can set breakpoints and create and save watch lists of variables to help you better understand what is going on inside your code. To manually reset the micro using the RESET button, move J14 to the USR position and select the appropriate Target Micro (see J6 on the next page)

In this example, the nanoEASE Debugger will control RESET functionality [J14] (necessary for debugging). However, power [J7] will come from the board, not the nanoEASE Debugger.

J5 & J6 - Target Micro Selector

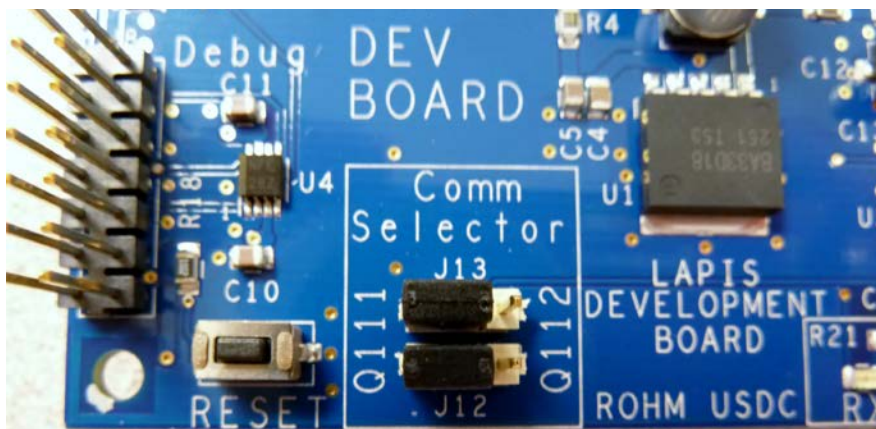
The debugger uses two dedicated pins to communicate and control the micros (TEST & RESET). Since the debugger should obviously only ever communicate with one micro at a time, jumpers J5 (TEST) and J6 (RESET) select which micro (Q111 or Q112) that the debugger is connected to. Note that to use the RESET button you must have the target micro selected and J14 set to USR.



Note: Both jumpers should select the same device.

In this example, the nanoEASE Debugger will control the Q112 micro.

J12 & J13 - UART COMM Selector



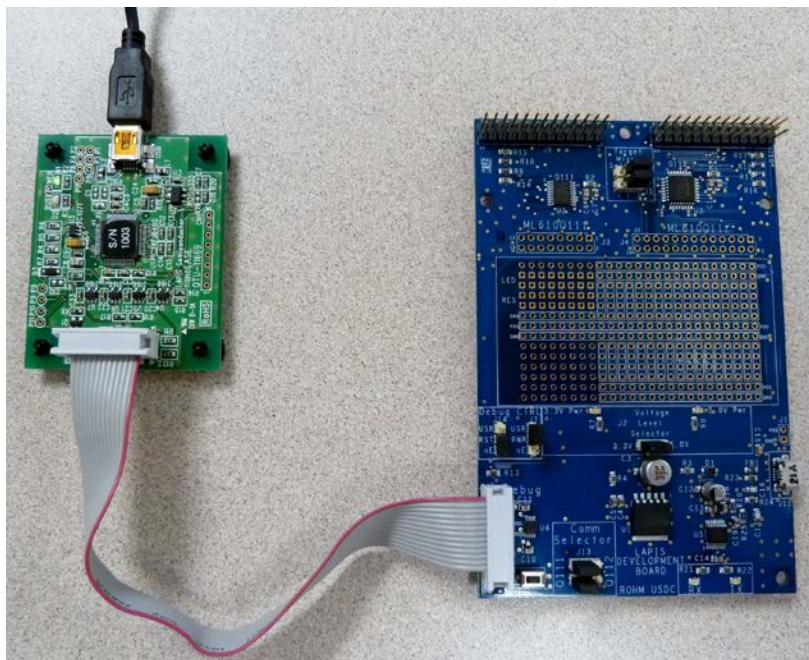
As mentioned previously, this LAPIS development board can connect a computer to the micro via the FTDI USB to UART translator IC to allow easy communications between the computer's USB

communication port and the micro's UART port. To select which micro you would like to communicate with use J12 (MCU-RX) and J13 (MCU-TX).

Please note that both jumpers should select the same device, but that these jumpers do not need to match the J5 & J6 target micros. In this example, the computer's USB port will connect to the Q111 micro.

nanoEASE Debugger Connection

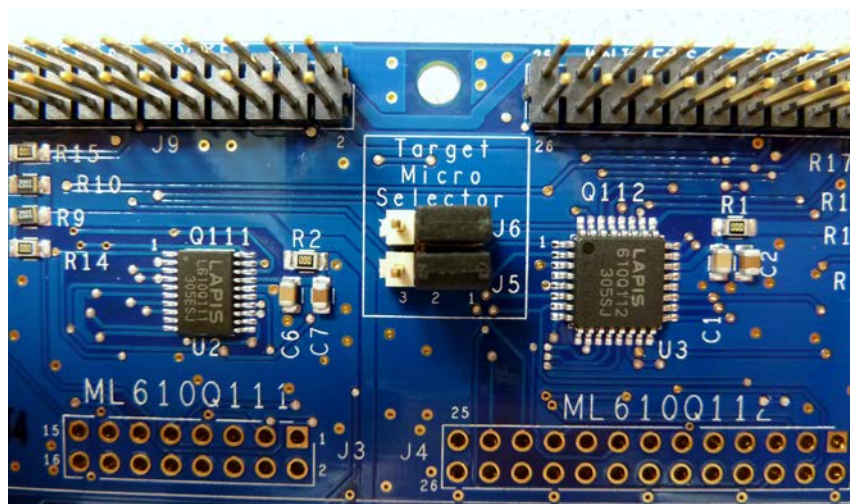
This LAPIS Development board has been designed to work with either the uEASE or the nanoEASE debuggers. The main difference between the two tools is the operating voltage. The uEASE power supplies 5.0V to the micros if desired (and selected by J7), while the nanoEASE provides 3.3V of power. Care should be taken to align Pin 1 of the debugger cable (Red Stripe) with Pin 1 on the J8 debugger port on the LAPIS board (see picture at right).



In this example, the LAPIS Development Board is connected to the nanoEASE Debugger.

MICROCONTROLLER GPIO AND UNIVERSAL SOCKETS

Both micros have all of their I/O pins routed down to the prototyping area [J3 & J4]. The Q111 section [J3] has 15 I/O pins (Pin 16 is the Ground connection), while the Q112 area [J4] has 25 I/O pins (Pin 26 is the Ground connection). In addition, both micros have a port for connecting to common pinouts – Universal Socket #1 [J9] for the Q111 micro and Universal Socket #2 [J10] for the Q112 micro.

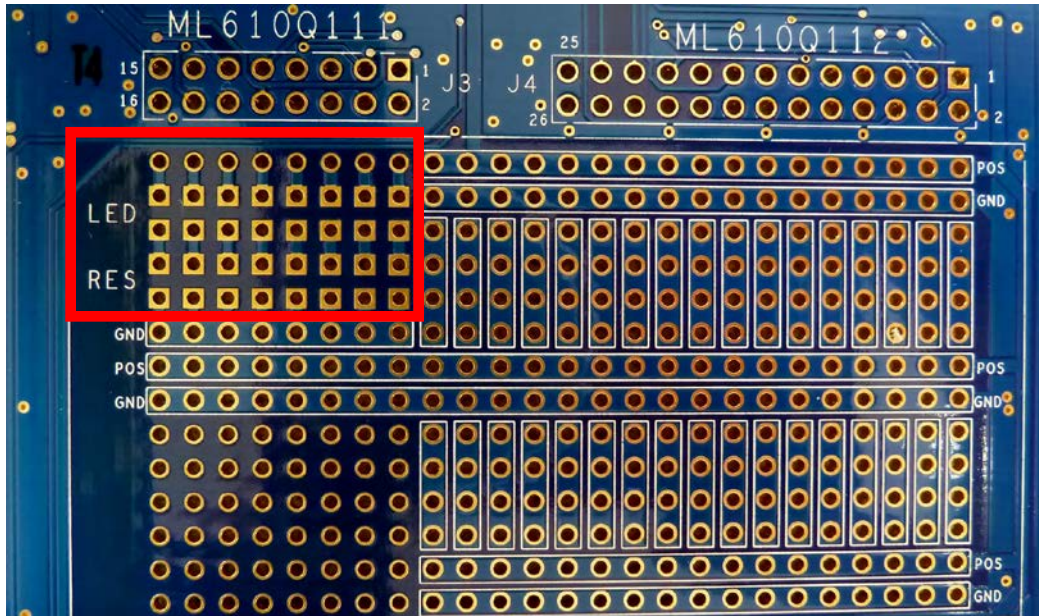


The pictures below show a sample Plug 'n Play board that connects to both universal sockets.

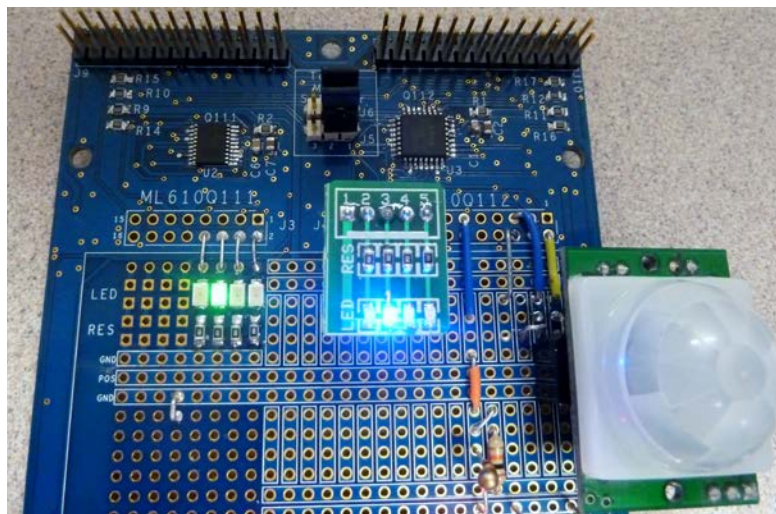


Prototyping Area

The LAPIS development boards have been designed with an easy-to-use prototyping area. This is a detailed picture of the prototyping area. Note that the square pads in the



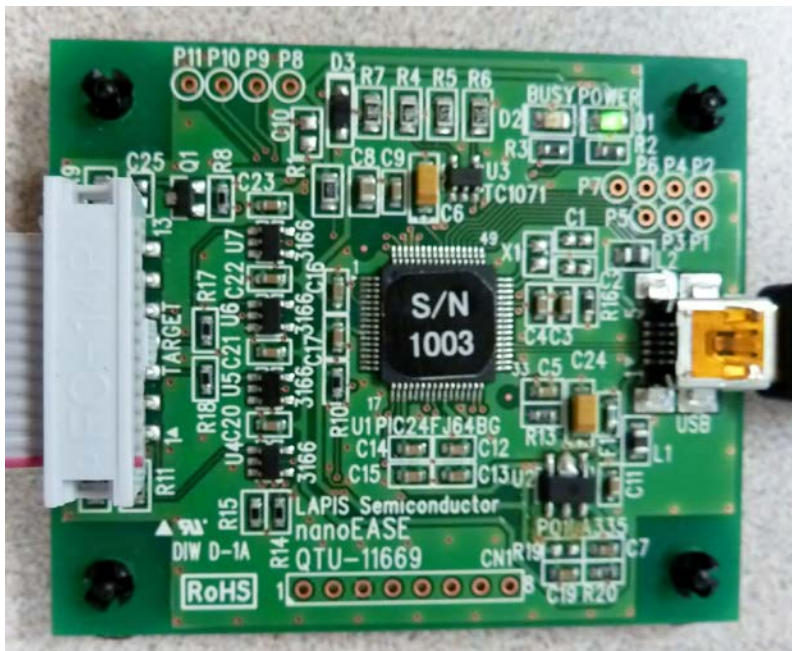
upper left (red box) are designed for either surface mount 0805 LEDs and resistors or through-hole LEDs and resistors. There are 8 prewired traces that can be easily connected to a desired I/O pin from either micro.



Example using the prototyping area.

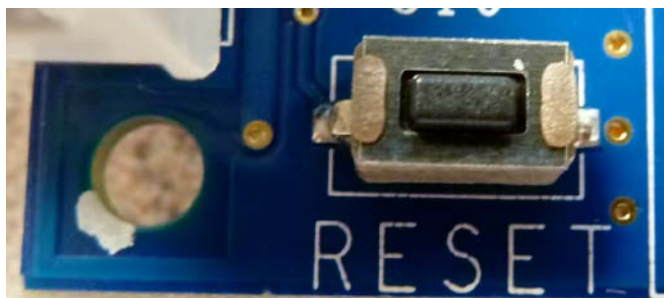
nanoEASE MICROCONTROLLER PROGRAMMER & DEBUGGER

The evaluation board includes a connector (J8) specifically designed to work with the low-cost, extremely useful LAPIS nanoEASE programmer and debugger. This versatile, powerful tool can program, develop and debug your custom applications in-circuit. It features an interface of 3.3V. In addition, since the microcontrollers on this evaluation board can be run at either



3.3V or 5.0V, we have designed in a 2-channel voltage level shifter for the RST & TST pins that allows the nanoEASE to communicate with the micro. It is an automatic system, so there is nothing the user needs to do to make it work. It simply ensures that the micro sees the appropriate voltage levels on these two pins during programming and debugging.

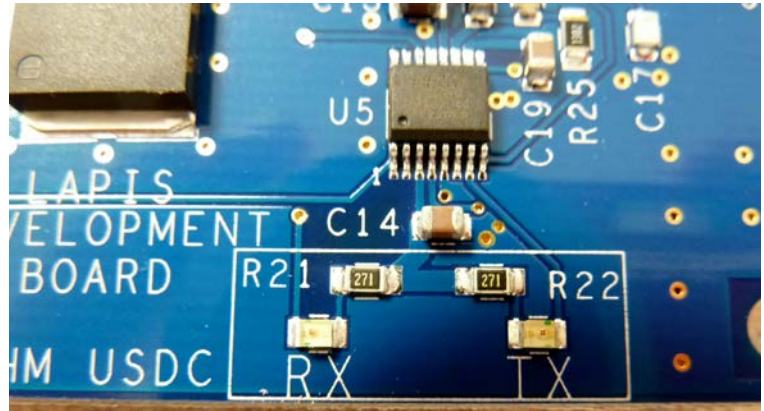
MICROCONTROLLER RESET BUTTON



The RESET button on the lower left corner of the LAPIS board (S1) resets the target microcontroller when pressed (provided J14 is set to 'USR').

USB TO SERIAL UART INTERFACE

The built-in USB to UART interface allows easy communication between a computer and the microcontrollers over USB using an FTDI FT230XS-R interface IC. This FTDI IC handles the USB to serial MCU UART translation and is fully compliant with USB 2.0 specifications.



USB Driver Support

Royalty-free Virtual COM Port (VCP) Drivers are available for:

- Windows 8 (32/64bit)
- Windows XP and XP 64bit
- Windows CE (4.2, 5.0 and 6.0)
- Windows 7 (32/64bit)
- Server 2003, XP, and Server 2008
- Mac OS-X
- Windows Vista and Vista 64bit
- Windows XP Embedded
- Linux 3.2 and greater
- Android

<http://www.ftdichip.com/Drivers/VCP.htm>

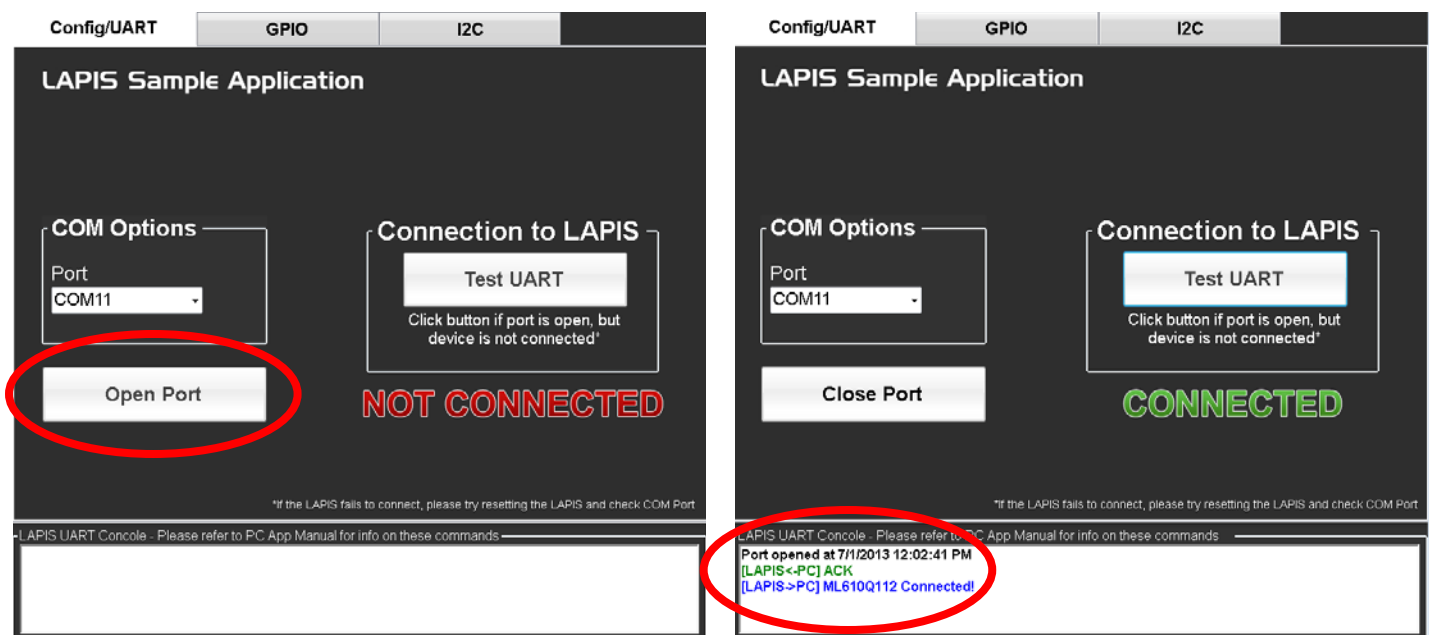
Using common development programming languages such as Python and C#, developers could create a simple interface program and send commands to the micro via USB. We have included a sample Graphical User Interface (GUI) to demonstrate this feature. **Please refer to the next section and/or the LAPIS GUI user manual for details.**

The internal MTP memory on the FT-230XS device can be re-programmed directly over USB if desired. No additional hardware, connections or programming voltages are required. The device is simply connected to the host computer in the same way that it would be for normal applications, and the FT_Prog utility is used to set the required options and program the device. The FT_Prog utility is provided free-of-charge from FTDI's website, and can be found at the link below. The user guide is also available at: http://www.ftdichip.com/Support/Utilities.htm#FT_Prog

LAPIS GRAPHICAL USER INTERFACE (GUI)

The custom GUI that is available for this LAPIS development board was designed to get you started using the LAPIS micros and to provide you with a tool that will allow you to control the microcontroller outputs from a computer. Please see the GUI User Manual for complete details. This section merely introduces you to this powerful and useful GUI tool.

Configuring Your Interface



1. First, ensure that you have downloaded and installed the FTDI drivers to enable your LAPIS development board to communicate with the LAPIS GUI.
2. Connect your LAPIS development board to a PC with the included microUSB cable.
3. Click on the LAPIS GUI to run it.
4. Select the COM port that plugs into the LAPIS board.
5. Select the appropriate Baud Rate based on your firmware specifications (the default is 9600 Baud).
6. Select your desired Parity, Stop Bits and Data Bits (the default is NONE-1-8).
7. Click the OPEN PORT button. You should get a confirmation in the console window at the bottom of the GUI.

Selecting the GPIO Tab

Now that the LAPIS development board is connected to your GUI, select the GPIO tab. This GUI currently allows you to control 8 of the GPIO pins from the Q112 micro. Over time, this GUI will be updated to include more features and options. But even as it currently stands, you will find it very useful in getting started.

The screenshot displays the GUI interface for configuring GPIO pins. The 'GPIO' tab is selected, indicated by a red arrow. The interface is divided into sections for Port B, Port C, and Port D. Port C pins C.0 through C.7 are listed, each with an input status of 'N/A' and an output status of 'Not Used'. A photograph of the development board shows the microcontroller and a connector with pins labeled C0 through C7. A UART console at the bottom shows the board is connected.

Port	Pin	Input	Output
Port C	C.0	Input <input checked="" type="radio"/> N/A	Output <input type="radio"/> Not Used
	C.1	Input <input checked="" type="radio"/> N/A	Output <input type="radio"/> Not Used
	C.2	Input <input checked="" type="radio"/> N/A	Output <input type="radio"/> Not Used
	C.3	Input <input checked="" type="radio"/> N/A	Output <input type="radio"/> Not Used
	C.4	Input <input checked="" type="radio"/> N/A	Output <input type="radio"/> Not Used
	C.5	Input <input checked="" type="radio"/> N/A	Output <input type="radio"/> Not Used
	C.6	Input <input checked="" type="radio"/> N/A	Output <input type="radio"/> Not Used
	C.7	Input <input checked="" type="radio"/> N/A	Output <input type="radio"/> Not Used

UART Console - Please refer to PC App Manual for info on these commands

```

Port opened at 7/1/2013 12:07:18 PM
[LAPIS<-PC] ACK
[LAPIS->PC] ML610Q112 Connected!

```

Controlling GPIO Outputs via GUI

Now you can begin controlling the outputs

The screenshot displays the GUI for controlling GPIO outputs. The 'GPIO' tab is active, and 'Port D' is selected. The 'Set all Outputs' button (1) is highlighted in pink, and the 'Set HIGH' button (2) is highlighted in blue. The output table shows the following states:

Pin	Input	Output
C.0	Not Used	High
C.1	Not Used	Low
C.2	Not Used	High
C.3	Not Used	Low
C.4	Not Used	High
C.5	Not Used	Low
C.6	Not Used	High
C.7	Not Used	Low

The console window (5) shows the following UART communication:

```

[LAPIS<-PC] OUC_00000000_01010101
[LAPIS->PC] OUC Received 01010101
[LAPIS<-PC] OUC_00000000_01010101
[LAPIS->PC] OUC Received 01010101
  
```

The diagram shows the header pins labeled C7, C5, C3, C1, C6, C4, C2, and C0.

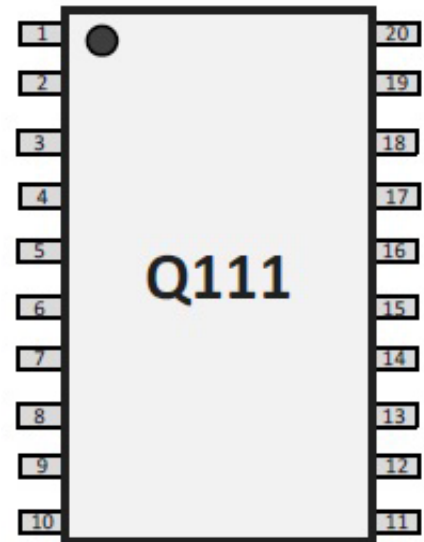
1. Press “Set all Output” to set all 8 I/O pins as outputs
2. Press “Set High” to set those 8 I/O pins ALL to a High state
3. Press the “HIGH” text in any of the outputs to toggle them to “LOW” state
4. Pressing “Set Output” sends the command to the micro that will actually set these pins to the desired state
5. The Console Window will show confirmation of the command sent from the PC to the LAPIS, as well as the data received and echoed back from the LAPIS’ micro to the PC
6. Press “Get Input” to read inputs

LAPIS Q111 PINOUT GUIDE

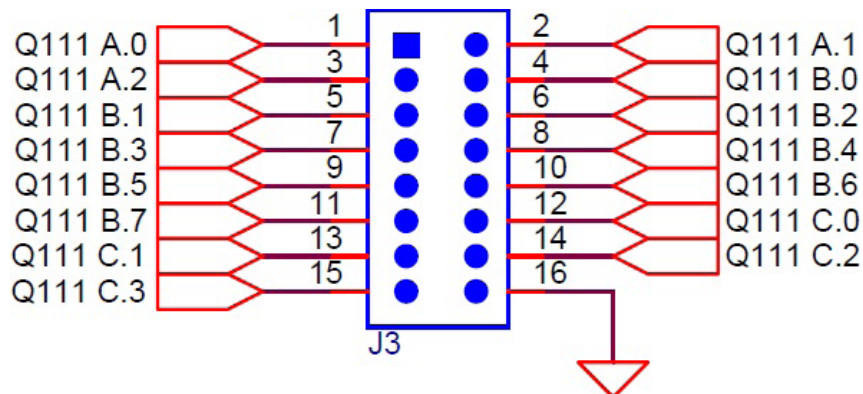
Both the Q111 & Q112 micros feature a rich array of peripherals. Each pin on the micros supports multiple possible functions. This diagram will help give an overview of the possibilities. Please see the Q111/Q112 User Manual for more details.

Pin Layout of the ML610Q111 TSSOP20 Package

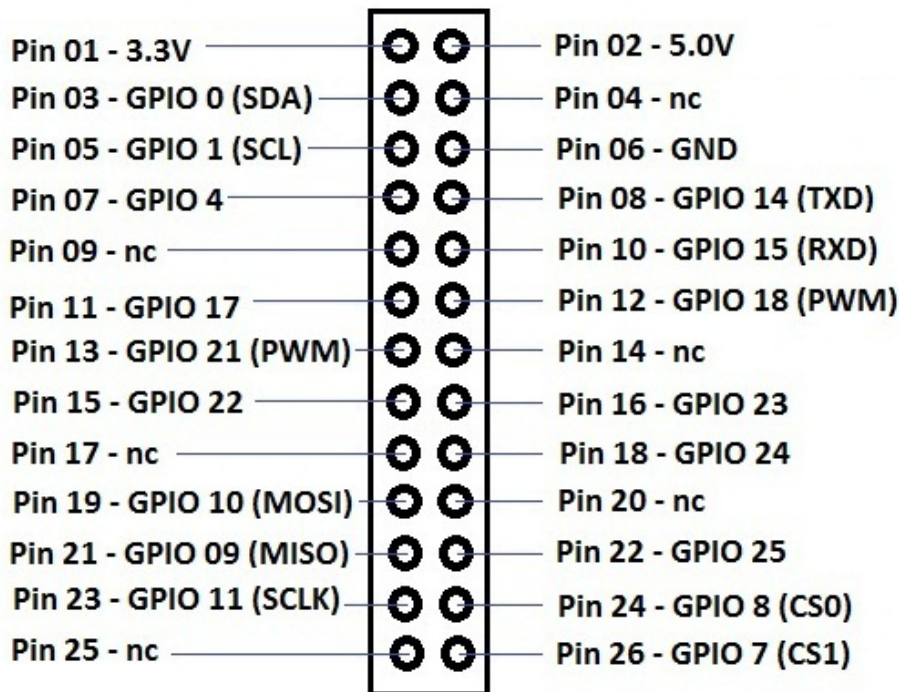
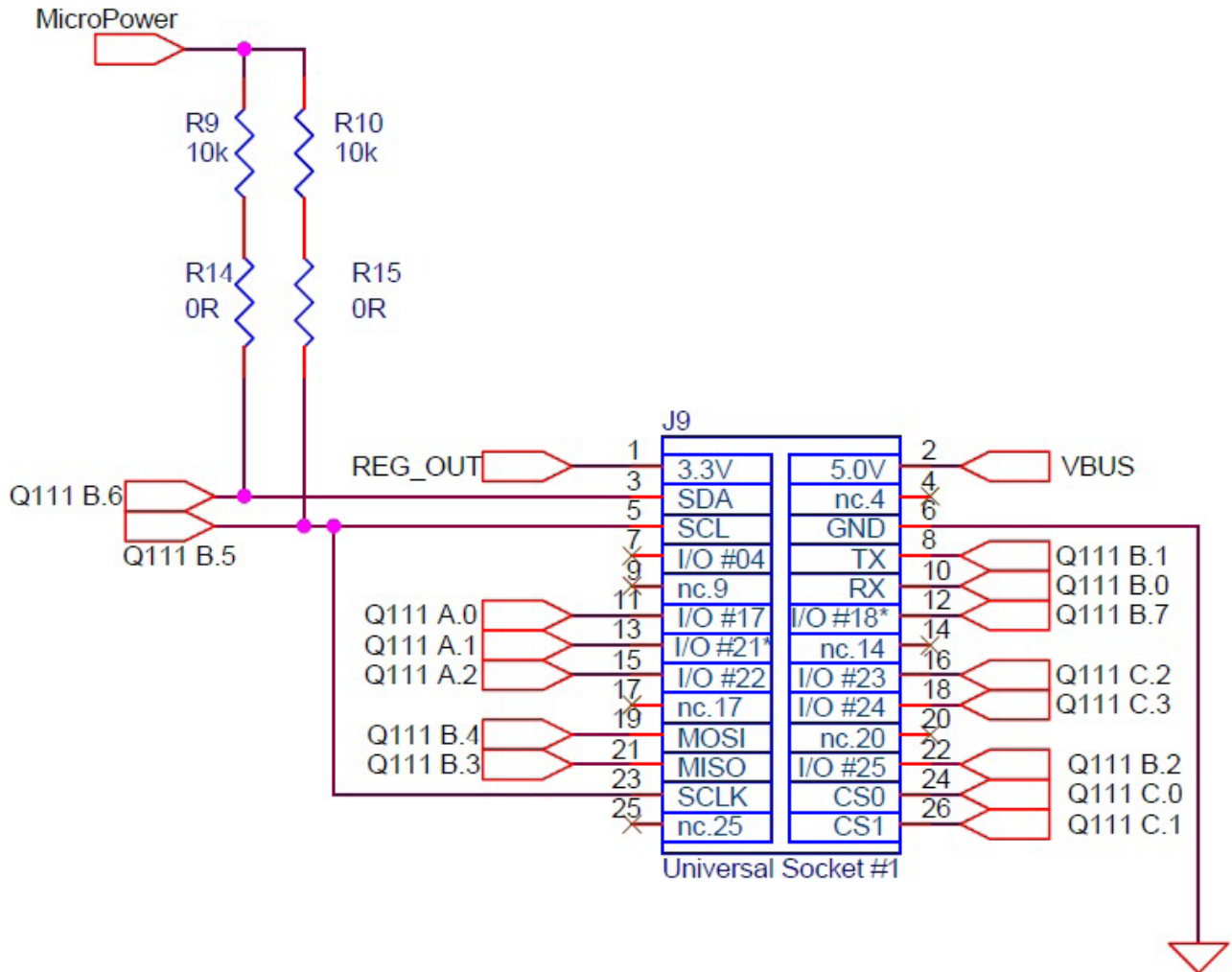
Pin 01	-	PC0	PWMF0	;TMR9OUT
Pin 02	-	RESET		
Pin 03	-	Test		
Pin 04	-	PB0	RXD0	;EXI4; AIN2 ;PWMC ;OUTCLK ;CMP1OUT
Pin 05	-	PB1	TXD0	;EXI5; AIN3 ;PWMD ;TXD1
Pin 06	-	PB2	EXI6	;PWME ;RXD1
Pin 07	-	PB3	EXI7	;SIN ;TXD1
Pin 08	-	PA2	EXI2	;PWME ;CLKIN ;COMPOOUT
Pin 09	-	TESTF		
Pin 10	-	PC3	TMFOUT	
Pin 11	-	PC2	PWMF2	
Pin 12	-	PA1	EXI1	;AIN1 ;CMP1P ;PWMD ;LSCLK ;TMFOUT
Pin 13	-	PB4	CMPOP	;SOUT ;TXD0 ;TXD1
Pin 14	-	PB5	CMPOM	;RXD0 ;SCK ;SCL ;PWMF2
Pin 15	-	PB6	AIN4	;CLKIN ;SDA ;PWMF1
Pin 16	-	VSS	(GROUND)	
Pin 17	-	VDD	(2.7-5.5V)	
Pin 18	-	PB7	AIN5	;RXD1 ;LSCLK ;PWMF0 ;PWMC
Pin 19	-	PA0	EXI0	;AIN0 ;PWMC ;OUTCLK ;TM9OUT
Pin 20	-	PC1	PWMF1	



J3 - All 15 I/O Pins of the Q111 Micro



J9 - Q111 Universal Socket #1 with Pinout - Compatible with the Raspberry Pi GPIO Layout



**Raspberry Pi®
GPIO Ports
(For reference only)**

LAPIS Q112 PINOUT GUIDE

Both the Q111 & Q112 micros feature a rich array of peripherals. Each pin on the micros supports multiple possible functions. This diagram will help give an overview of the possibilities. Please see the Q111/Q112 User Manual for more details.

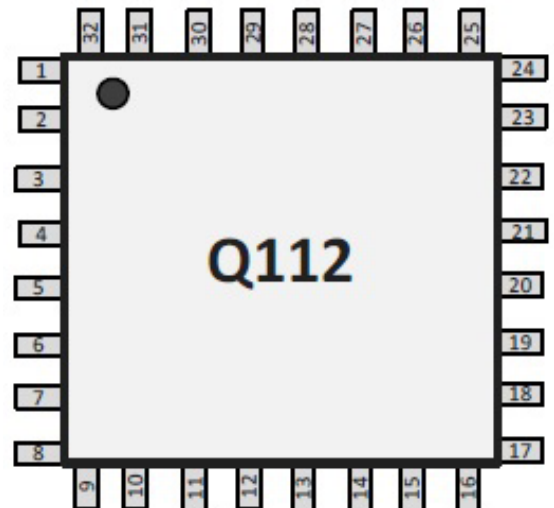
Pin Layout of the ML610Q112 LQFP32 Package

Pin 01 - *Test*
 Pin 02 - PD1
 Pin 03 - PB0 RXD0 ;EXI4; AIN2 ;PWMC ;OUTCLK ;CMP1OUT
 Pin 04 - *n.c.*
 Pin 05 - PB1 TXD0 ;EXI5; AIN3 ;PWMD ;TXD1
 Pin 06 - PB2 EXI6 ;PWME ;RXD1
 Pin 07 - PB3 EXI7 ;SIN ;TXD1
 Pin 08 - PA2 EXI2 ;PWME ;CLKIN ;COMPOOUT

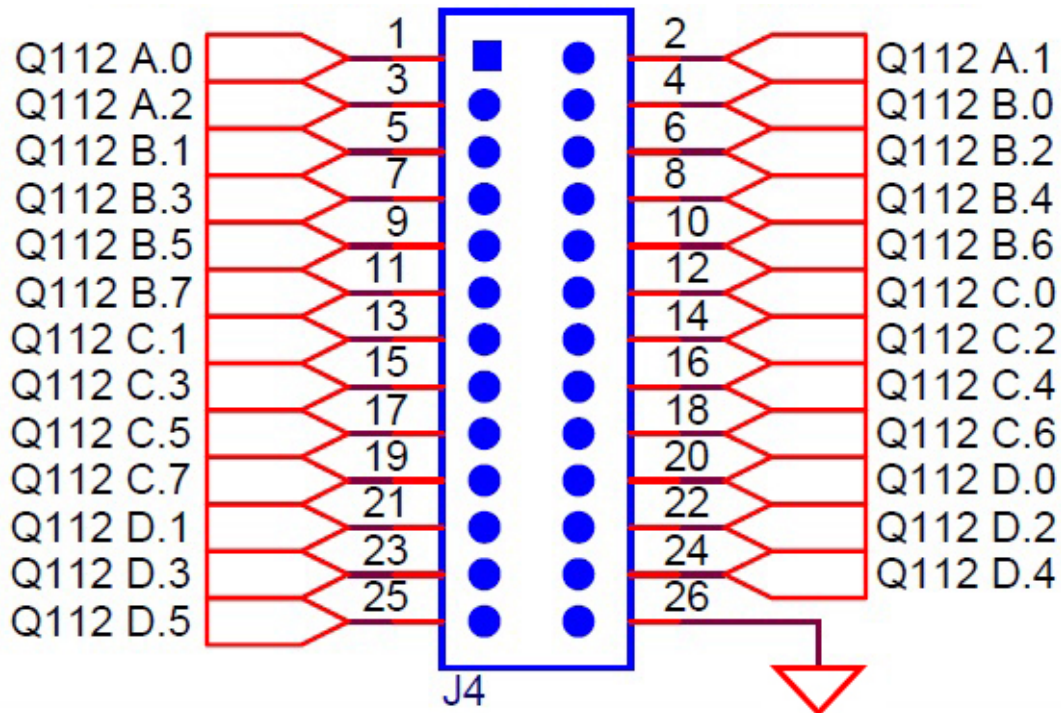
Pin 09 - *TESTF*
 Pin 10 - PD2
 Pin 11 - PC3 TMFOUT
 Pin 12 - PD3
 Pin 13 - PD4
 Pin 14 - PC2 PWMF2
 Pin 15 - PD5
 Pin 16 - PA1 EXI1 ;AIN1 ;CMP1P ;PWMD ;LSCLK ;TMFOUT

Pin 17 - PB4 CMP0P ;SOUT ;TXD0 ;TXD1
 Pin 18 - PB5 CMP0M ;RXD0 ;SCK ;SCL ;PWMF2
 Pin 19 - PB6 AIN4 ;CLKIN ;SDA ;PWMF1
 Pin 20 - *n.c.*
 Pin 21 - VSS (GROUND)
 Pin 22 - VDD (2.7-5.5V)
 Pin 23 - PC7 AIN7
 Pin 24 - PB7 AIN5 ;RXD1 ;LSCLK ;PWMF0 ;PWMC

Pin 25 - PA0 EXI0 ;AIN0 ;PWMC ;OUTCLK ;TM9OUT
 Pin 26 - PC6 AIN6
 Pin 27 - PC1 PWMF1
 Pin 28 - PC5 SDA
 Pin 29 - PC4 SCL
 Pin 30 - PC0 PWMF0 ;TMR9OUT
 Pin 31 - PD0
 Pin 32 - *RESET*

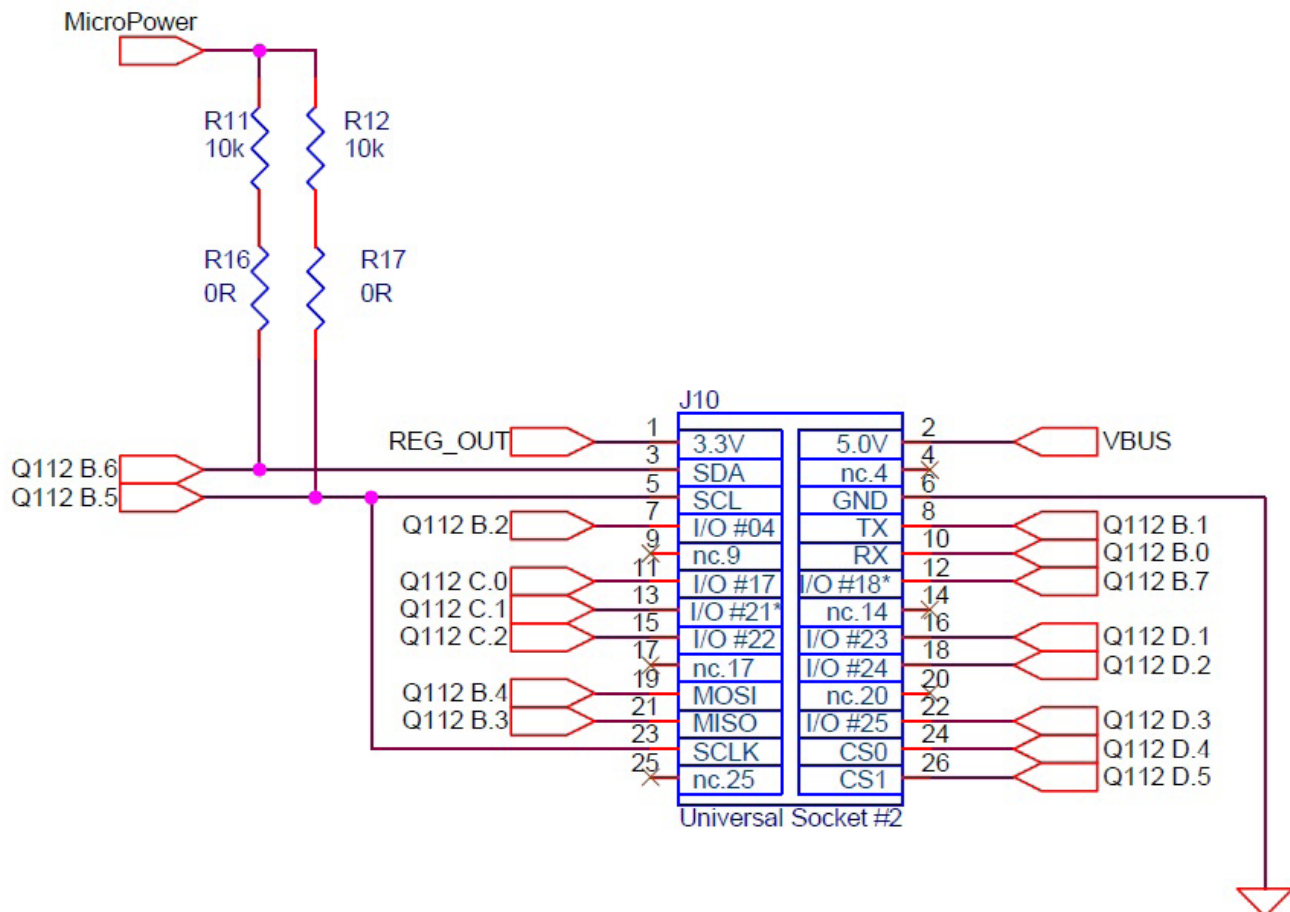


J4 - All 25 I/O Pins of the Q112 Micro



J10 - Q112 Universal Socket #2 with Pinout -

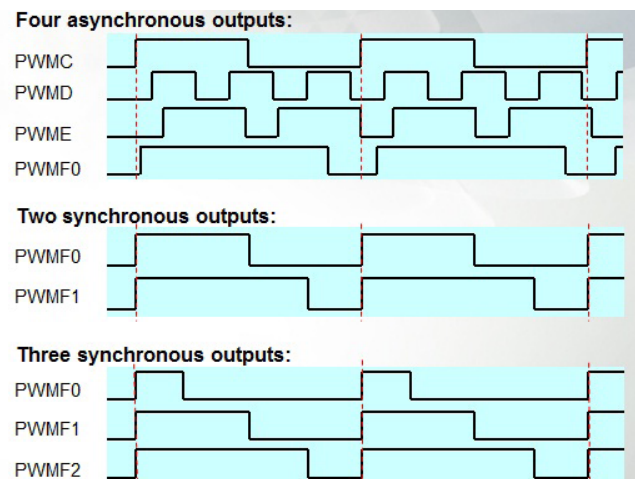
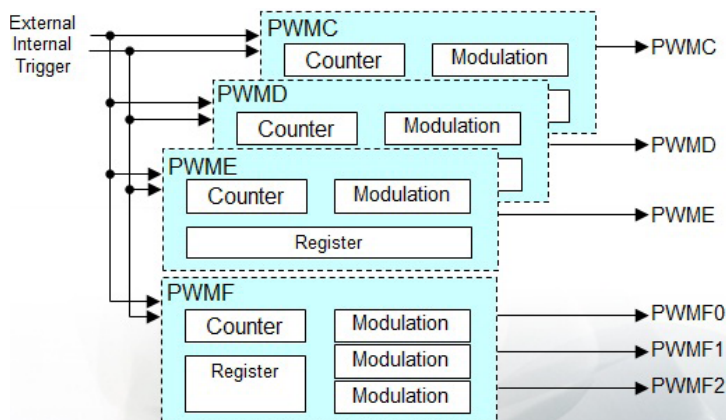
Compatible with the Raspberry Pi® GPIO Layout



Q111 & Q112 PWM Configuration Options

ML610Q111 / Q112 Specifications

- 16bit PWM x 4ch (3 synchronous ports)
 - » 16bit for all channels
 - » PWMC, PWMD, PWME have only one digital output
 - » PWMF has three synchronous digital outputs
 - » Selectable duty ratio on each of the three PWMF ports
 - » All interrupt requests of PWM are generated by period duty, and period or duty
 - » Trigger by external/internal hardware or software to start/stop PWM
- PWM Configuration
 - » In the case of four asynchronous outputs use PWMC, PWMD, PWME and PWMF0 (Ex. for a double LED brightness controller)
 - » In the case of two synchronous outputs use PWMF0 and PWMF1 (Ex. for a single LED brightness controller)
 - » In the case of three synchronous outputs use PWMF0, PWMF1 and PWMF2 (Ex. for an RGB LED controller)

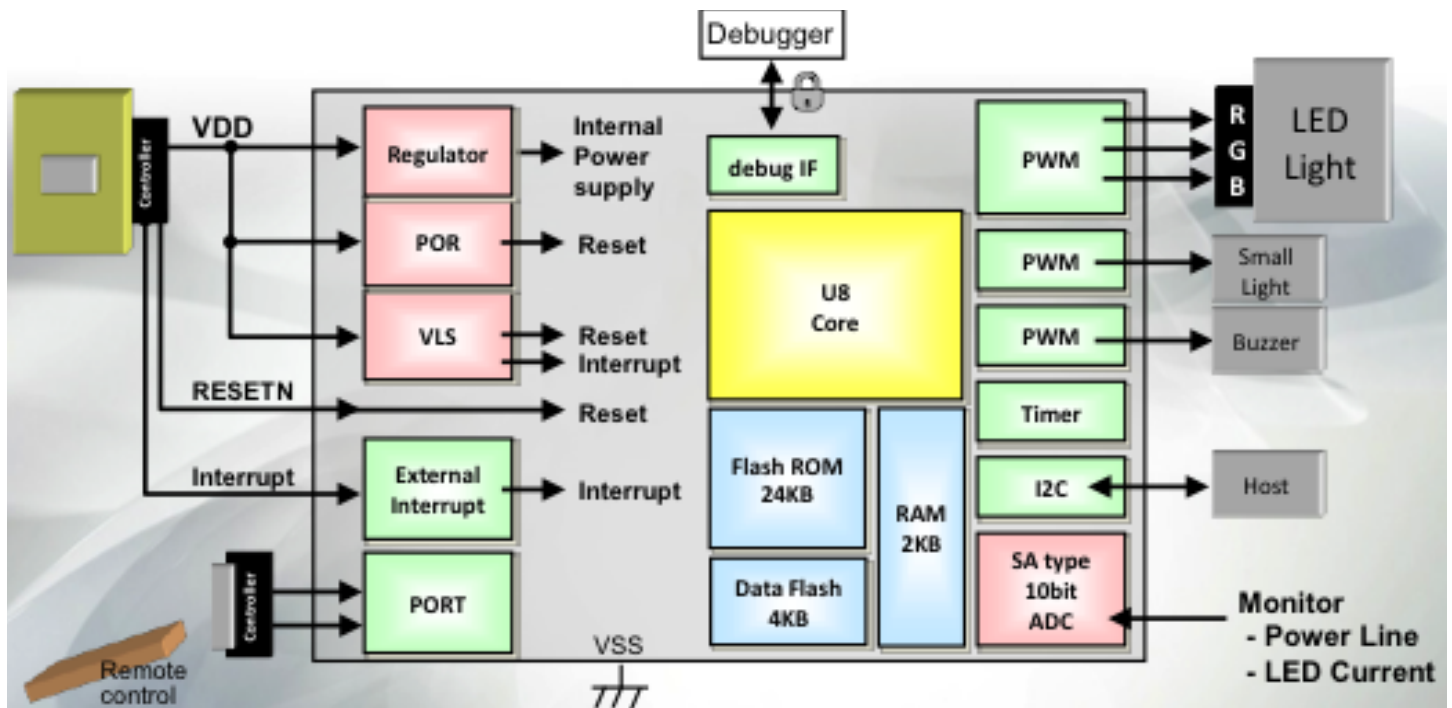


Application Example for Q111 & Q112 Micros

ML610Q111/Q112 Application

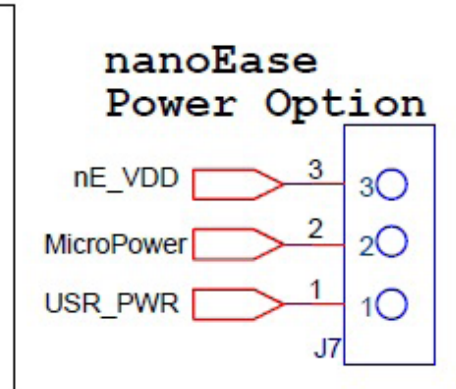
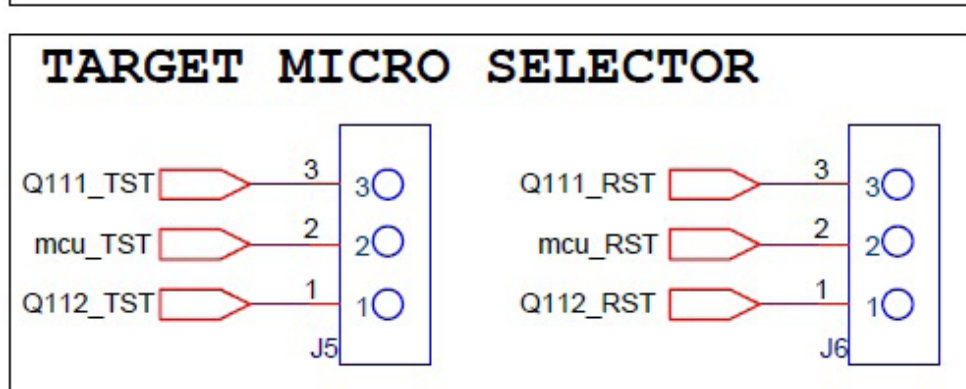
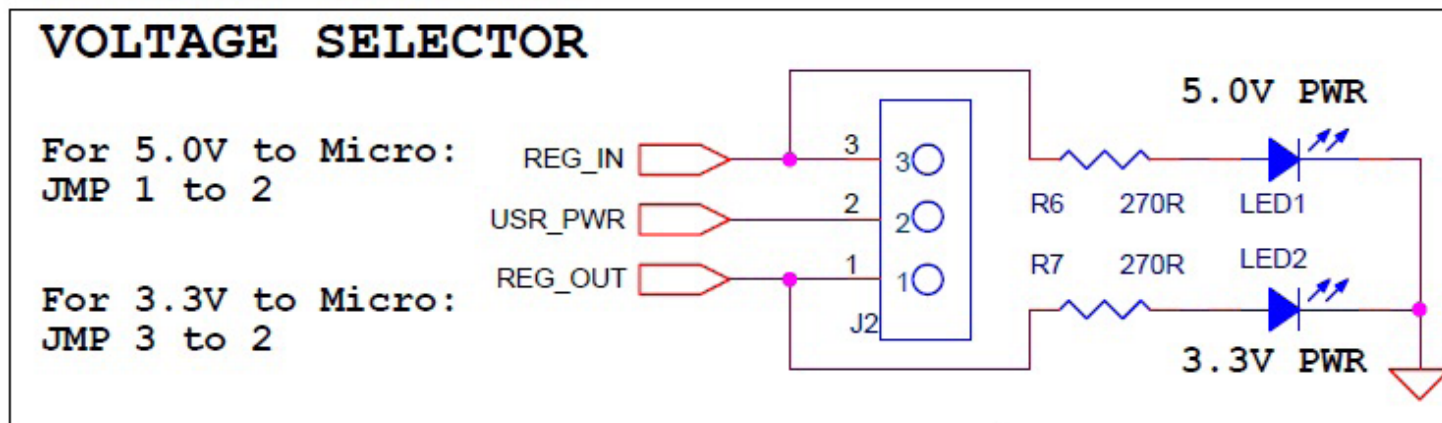
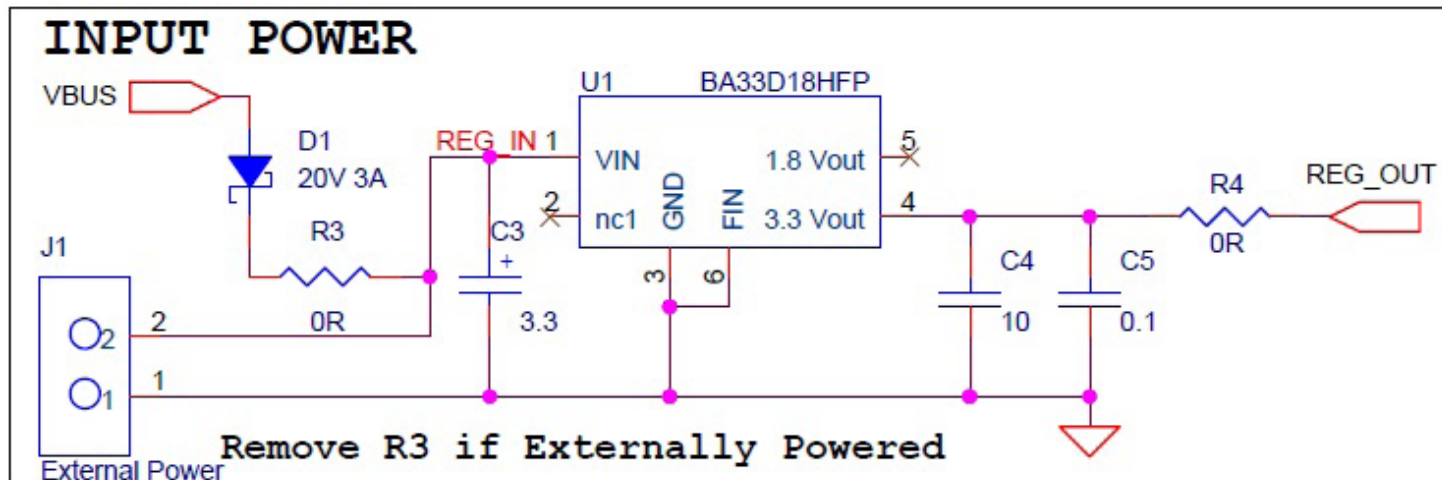
Package Type : 20pin TSSOP

Peripherals : Synchronous PWM (x3), PWM (x2), I²C interface, External Interrupt, General Purpose Port and ADC (x1)

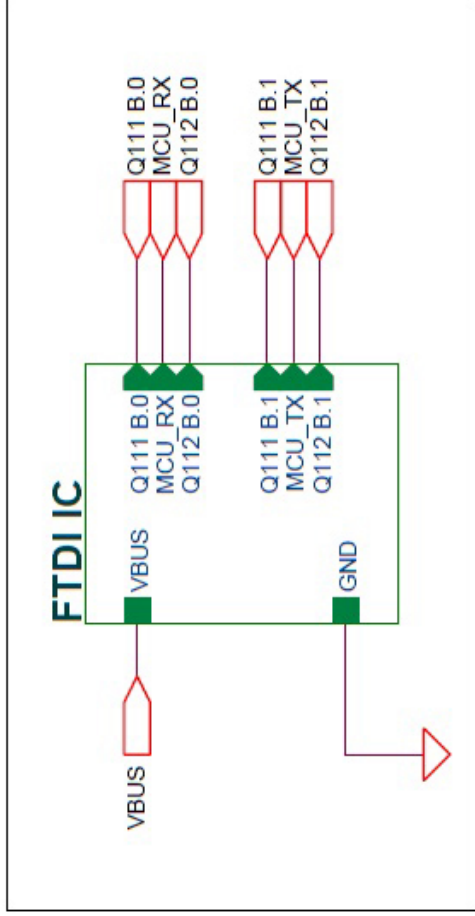
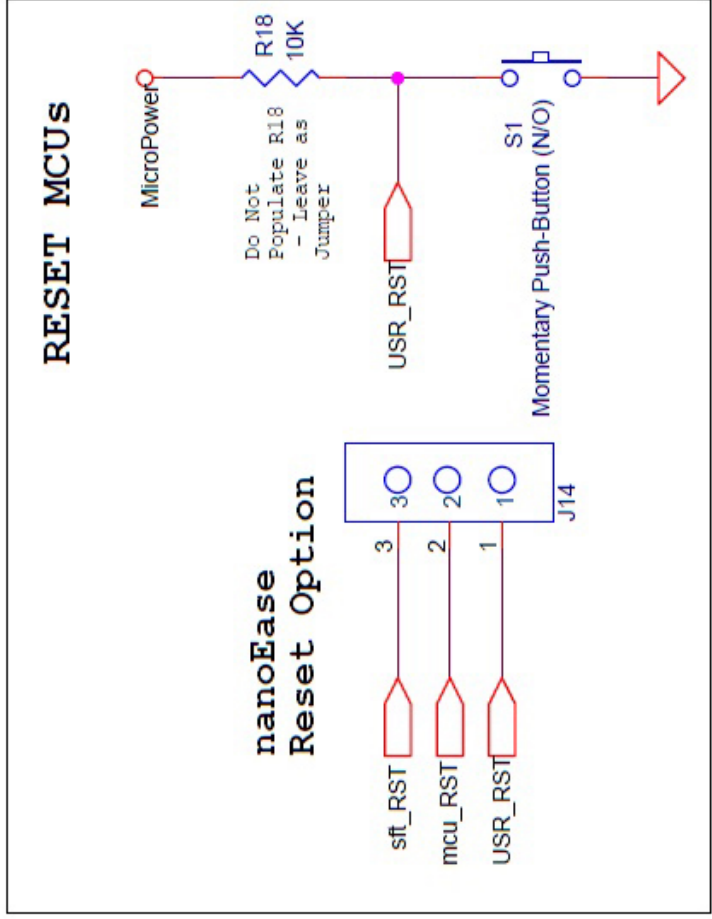
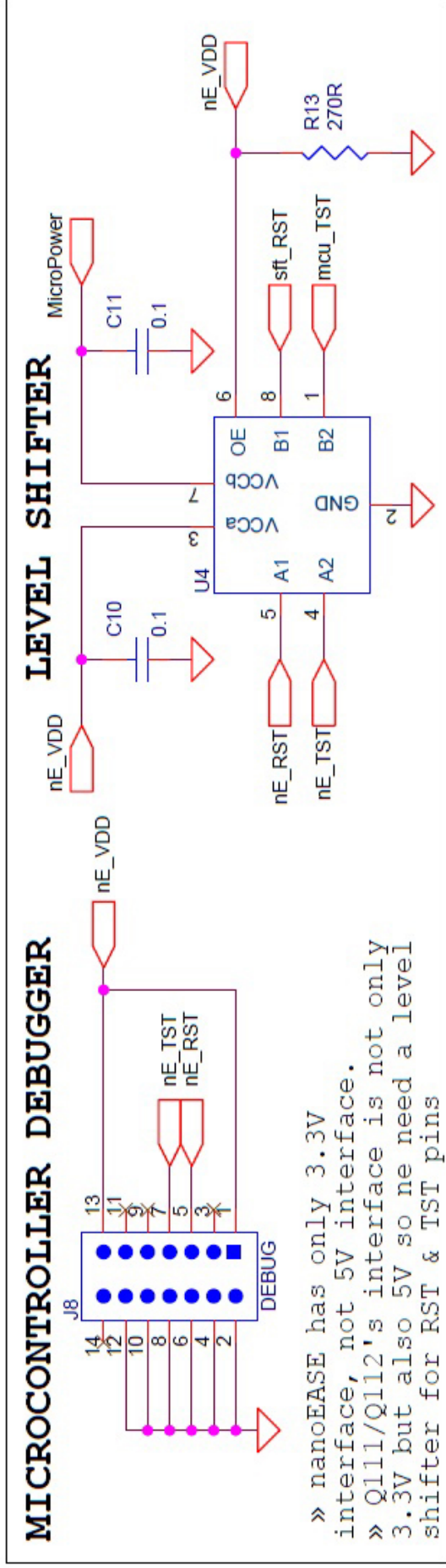


LAPIS DEVELOPMENT BOARD SCHEMATICS

Input Power Block

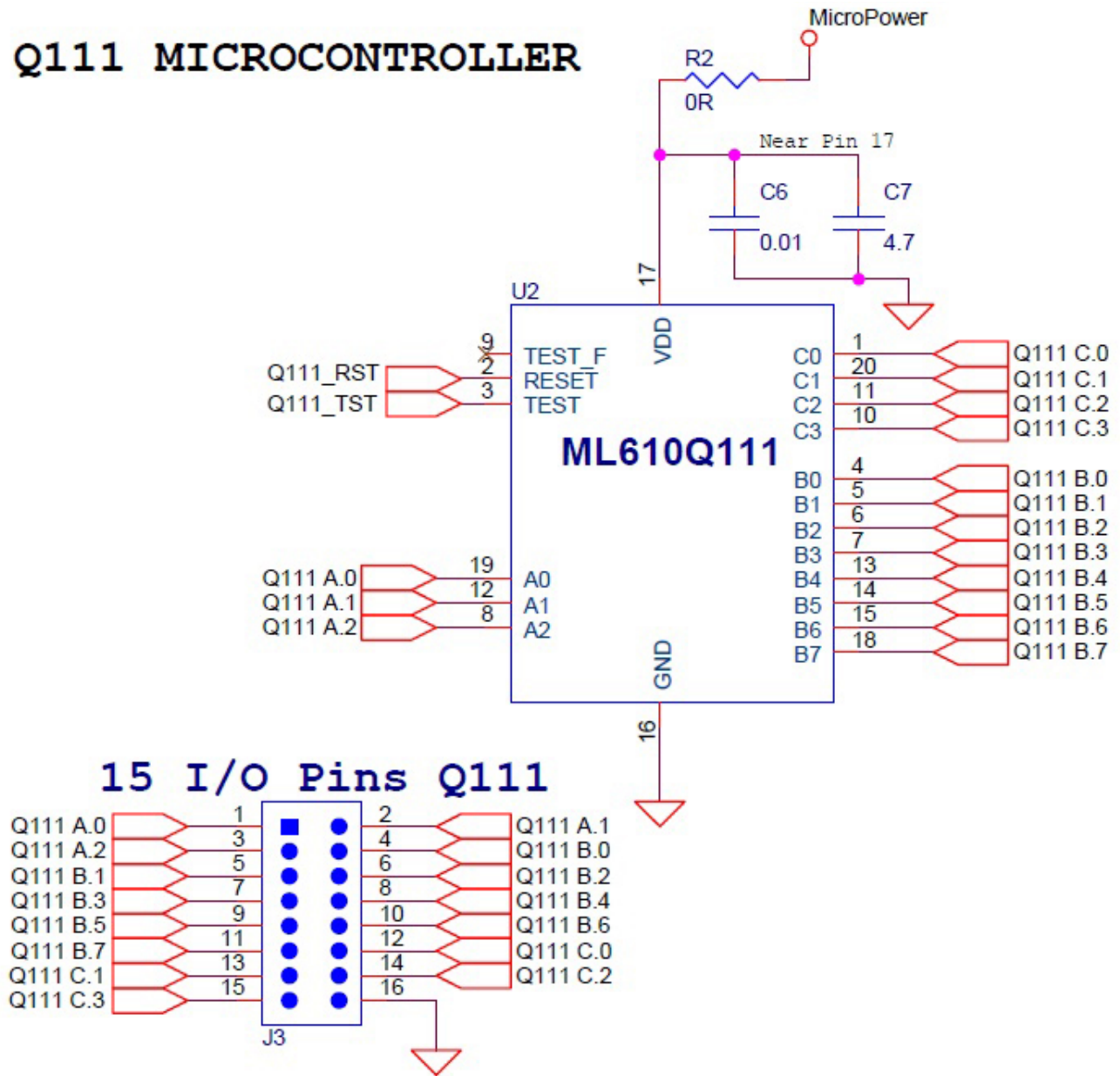


nanoEASE Block



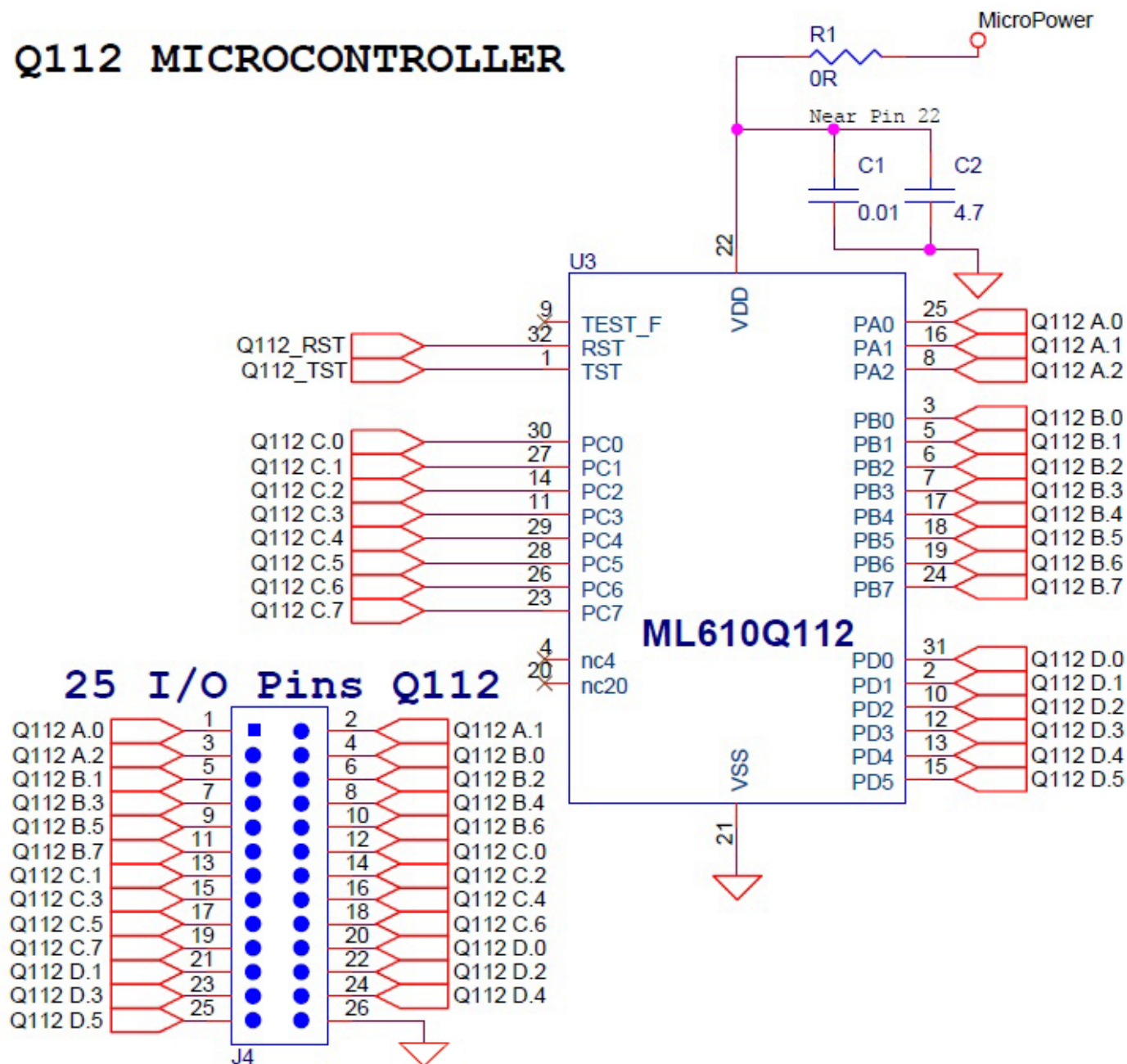
Q111 Micro Block

Q111 MICROCONTROLLER



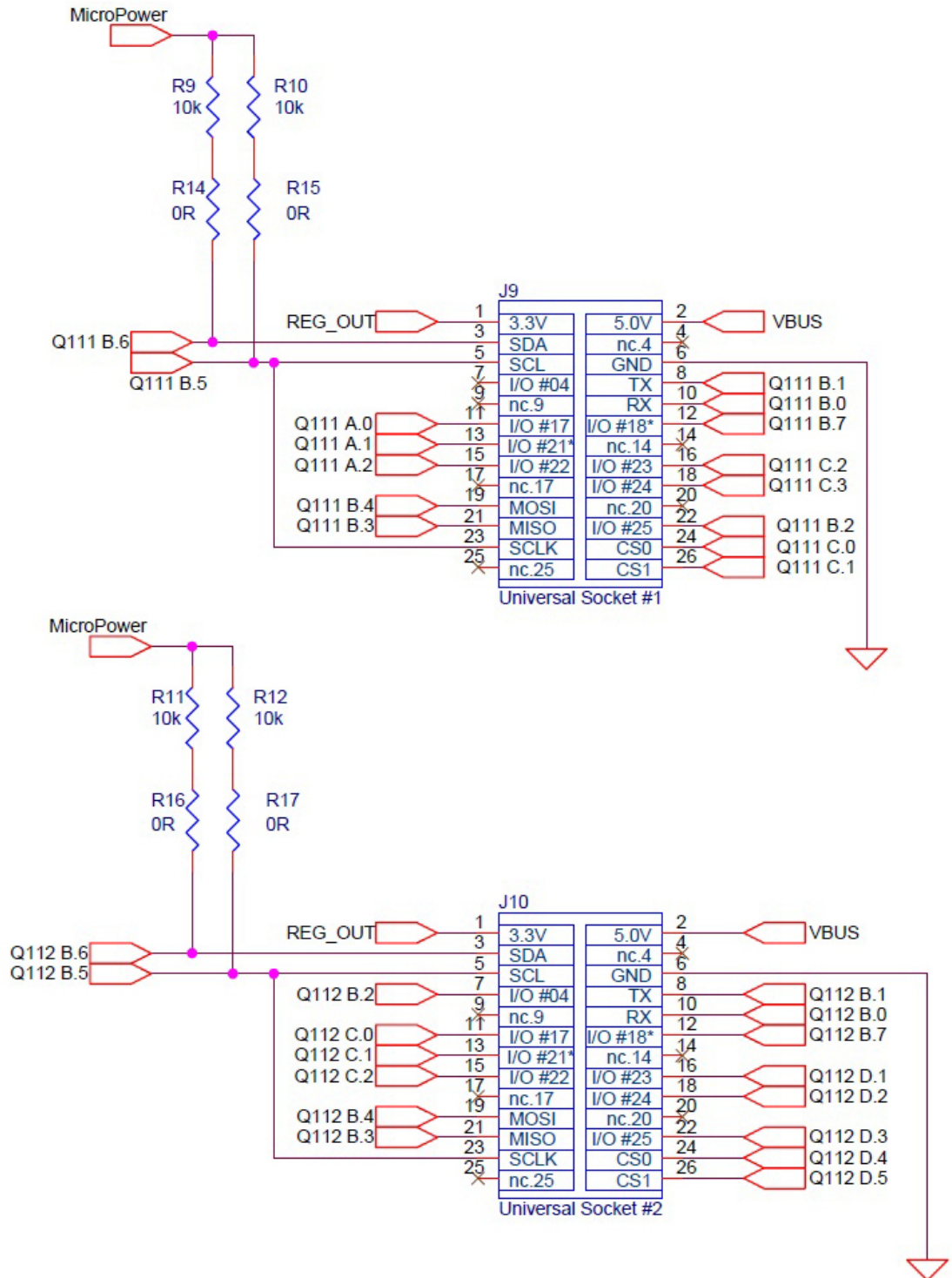
Q112 Micro Block

Q112 MICROCONTROLLER

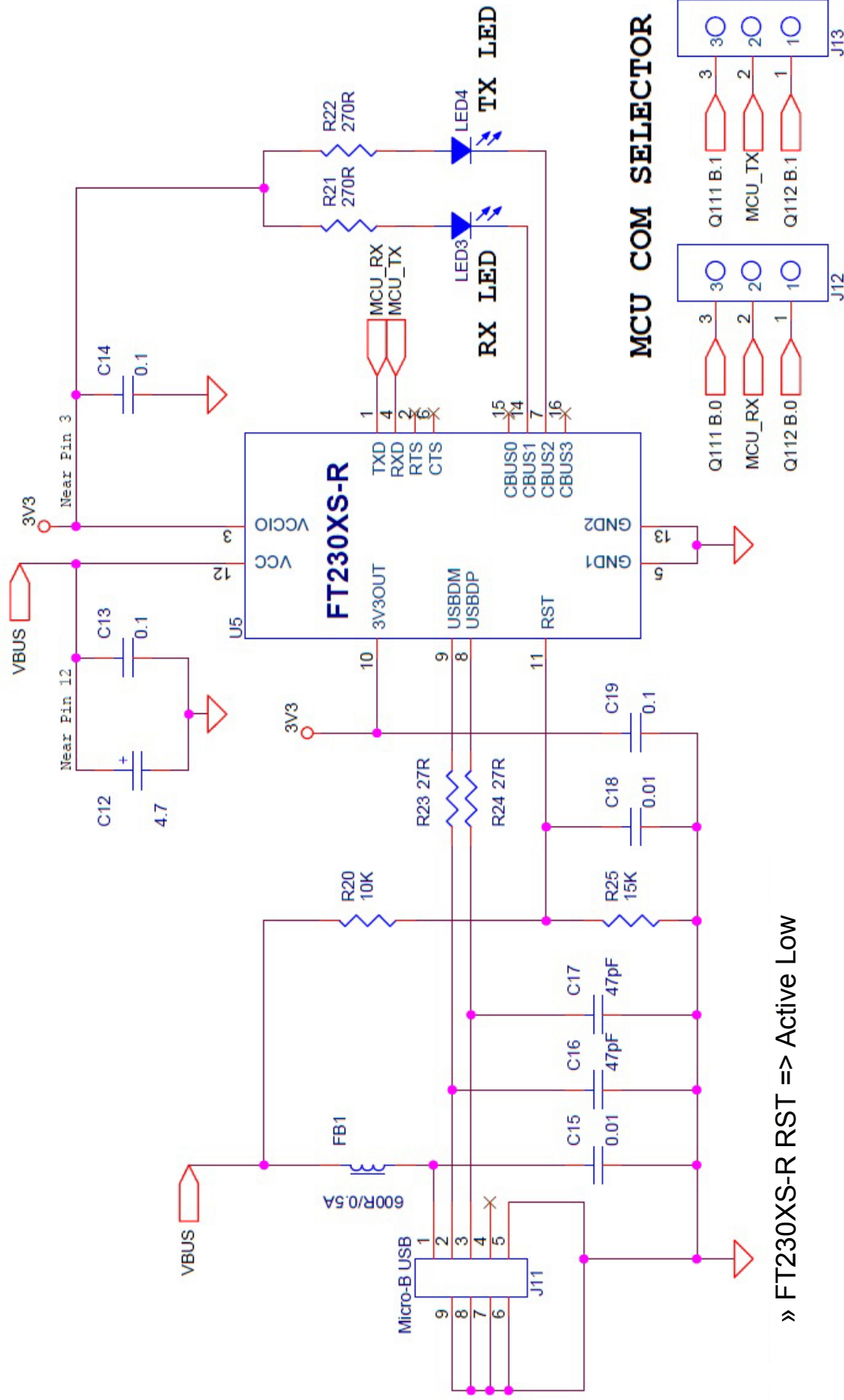


Universal Sockets Block

Raspberry Pi Compatible UNIVERSAL SOCKETS



USB to UART Circuit Block



» FT230XS-R RST => Active Low

The MTP memory on this FT-230XS-R device can be programmed directly over USB using the FT Prog utility to set the required options and program it. See: www.ftchip.com/support/utilities.html#FT_Prog

LAPIS DEVELOPMENT BOARD OF MATERIAL (B.O.M)

nanoEASE Block

LAPIS Development Board

Circuit Schematic Rev. 1.9

Last Updated: April 10th, 2013 (17:30)

#	REF	VALUE	FOOTPRINT	MFR	MFR P/N	Digi-Key P/N
1	J2	Voltage Selector	Through-hole - 1x3	Samtec	HTSW-103-07-G-S	HTSW-103-07-G-S-ND
2	J5	Target Micro Selector - TST				
3	J6	Target Micro Selector - RST				
4	J7	nanoEase PWR Opt Selector				
5	J12	MCU Com Selector - RX				
6	J13	MCU Com Selector - TX				
7	J14	nanoEase RST Opt Selector				
8	J8	Debugger	Through-hole - 2x7	Samtec	TSW-150-07-L-D	SAM1030-50-ND
9	J9	Universal Socket #1	Through-hole - 2x13			
10	J10	Universal Socket #2	Through-hole - 2x13			
11	J11	SMD - micro USB	Surface Mount - microUSB	Hirose	ZX62R-B-5P	H11574CT-ND
12	J1	External Power	xxx	xxx	xxx	xxx
13	J3	15 I/O Pins Q111	xxx	xxx	xxx	xxx
14	J4	25 I/O pins Q112	xxx	xxx	xxx	xxx
15	U1	3.3v LDO Reg (USB 5V to 3.3v, etc.)	Surface Mount - HRP5 (5-pins plus tab)	ROHM	BA33D18HFP	BA33D18HFPDKR-ND
16	U2	Q111 - 5V, 8-Bit, 20-pin LAPIS Micro	Surface Mount - TSSOP-20 (20-pin)	ROHM	ML610Q111	LAPIS-SAMPLES
17	U3	Q112 - 5V, 8-Bit, 20-pin LAPIS Micro	Surface Mount - LQFP-32 (32-pin)	ROHM	ML610Q112	LAPIS-SAMPLES
18	U4	Voltage Level Shifter (2-bit)	Surface Mount - DCT (8-pin)	Texas Instruments	TXS0102DCTR	296-21978-1-ND
19	U5	USB-to-UART Translator	Surface Mount - SSOP (16-pin)	FTDI	FT230XS-R	768-1135-1-ND
20	S1	Micro's Reset Button (Normally Open)	Surface Mount (2-pad)	E-Switch	TL3302AF180QG	EG2528DKR-ND
21	LED1	RED - 5V PWR	SMD 0805 (2012 Metric)	ROHM	SML-210VTT86	511-1289-1-ND
22	LED2	RED - 3.3V PWR				
23	LED3	GREEN - RX Indicator				
24	LED4	YELLOW - TX Indicator				
25	D1	Schottky Diode Input Protection - 20V 3A	Surface Mount - 123SOD (2-pads)	ROHM	RB051M-2YTR	RB051M-2YCT-ND
26	FB1	Ferrite Bead (600R/0.5A)	SMD 0805 (2012 Metric)	Laird	MI0805K601R-10	240-2390-1-ND

#	REF	VALUE	FOOTPRINT	MFR	MFR P/N	Digi-Key P/N
27	R1	0R	SMD 0805 (2012 Metric)	ROHM	MCR10EZPJ000	RHM0.0ARCT-ND
28	R2					
29	R3					
30	R4					
31	R14					
32	R15					
33	R16					
34	R17	27R	SMD 0805 (2012 Metric)	ROHM	MCR10EZPF270 2	RHM27.0KCRCT-ND
35	R23					
36	R24					
37	R6	270R	SMD 0805 (2012 Metric)	ROHM	MCR10EZPJ271	RHM270ACT-ND
38	R7					
39	R13					
40	R21					
41	R22					
42	R9	10k	SMD 0805 (2012 Metric)	ROHM	MCR10EZPF100 2	RHM10.0KCRCT-ND
43	R10					
44	R11					
45	R12					
46	R20					
47	R18					
48	R25					
49	C16	Cer - 47pF	SMD 0805 (2012 Metric)	Kemet	C0805C470J1G ACTU	399-8079-1-ND
50	C17					
51	C1	Cer - 0.01 μ F	SMD 0805 (2012 Metric)	TDK Corporation	C2012X7R2E10 3K	445-2280-1-ND
52	C6					
53	C15					
54	C18					
55	C5	Cer - 0.1 μ F	SMD 0805 (2012 Metric)	TDK Corporation	C2012X7R2A10 4K	445-1418-1-ND
56	C10					
57	C11					
58	C13					
59	C14					
60	C19	Polarized - 3.3 μ F, 50V	SMD Alum Cap (660x773)	Panasonic	EEE-1HA3R3NP	PCE4303DKR-ND
61	C3					
62	C12					
63	C2	Cer - 4.7 μ F	SMD 0805 (2012 Metric)	TDK Corporation	C2012Y5V1A47 5Z/1.25	445-1370-1-ND
64	C7					
65	C4	Cer - 10 μ F	SMD 0805 (2012 Metric)	TDK Corporation	C2012X5R1C10 6K085AC	445-7644-6-ND
66	Rubber Feet	5 per board	Stick-on Rubber Feet	3M	SJ-5003 (white)	SJ5003-9-ND
67	Chiclets	7 per board	Chicle Jumpers with Tab	Sullins Connector Solutions	NPC02SXON-RC	S9341-ND

NOTES

NOTES

For technical questions or assistance on the LAPIS Development Board, please contact us at:

Carl Schell
Applications Engineer - US Design Center
ROHM Semiconductor USA
Office: 248-348-9920 ext. 11215
Cell: 313-414-6481

26800 Meadowbrook Rd Suite 120
Novi, MI 48377-3540
cschell@rohmsemiconductor.com



Raspberry Pi® is a trademark of the Raspberry Pi Foundation. Other listed names and brands are trademarks or registered trademarks of their respective owners.

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [Power Management IC Development Tools](#) category:

Click to view products by [ROHM](#) manufacturer:

Other Similar products are found below :

[EVAL-ADM1168LQEBZ](#) [EVB-EP5348UI](#) [MIC23451-AAAYFL EV](#) [MIC5281YMME EV](#) [DA9063-EVAL](#) [ADP122-3.3-EVALZ](#) [ADP130-0.8-EVALZ](#) [ADP130-1.2-EVALZ](#) [ADP130-1.5-EVALZ](#) [ADP130-1.8-EVALZ](#) [ADP1712-3.3-EVALZ](#) [ADP1714-3.3-EVALZ](#) [ADP1715-3.3-EVALZ](#) [ADP1716-2.5-EVALZ](#) [ADP1740-1.5-EVALZ](#) [ADP1752-1.5-EVALZ](#) [ADP1828LC-EVALZ](#) [ADP1870-0.3-EVALZ](#) [ADP1871-0.6-EVALZ](#) [ADP1873-0.6-EVALZ](#) [ADP1874-0.3-EVALZ](#) [ADP1882-1.0-EVALZ](#) [ADP199CB-EVALZ](#) [ADP2102-1.25-EVALZ](#) [ADP2102-1.875EVALZ](#) [ADP2102-1.8-EVALZ](#) [ADP2102-2-EVALZ](#) [ADP2102-3-EVALZ](#) [ADP2102-4-EVALZ](#) [ADP2106-1.8-EVALZ](#) [ADP2147CB-110EVALZ](#) [AS3606-DB](#) [BQ24010EVM](#) [BQ24075TEVM](#) [BQ24155EVM](#) [BQ24157EVM-697](#) [BQ24160EVM-742](#) [BQ24296MEVM-655](#) [BQ25010EVM](#) [BQ3055EVM](#) [NCV891330PD50GEVB](#) [ISLUSBI2CKIT1Z](#) [LM2744EVAL](#) [LM2854EVAL](#) [LM3658SD-AEV/NOPB](#) [LM3658SDEV/NOPB](#) [LM3691TL-1.8EV/NOPB](#) [LM4510SDEV/NOPB](#) [LM5033SD-EVAL](#) [LP38512TS-1.8EV](#)