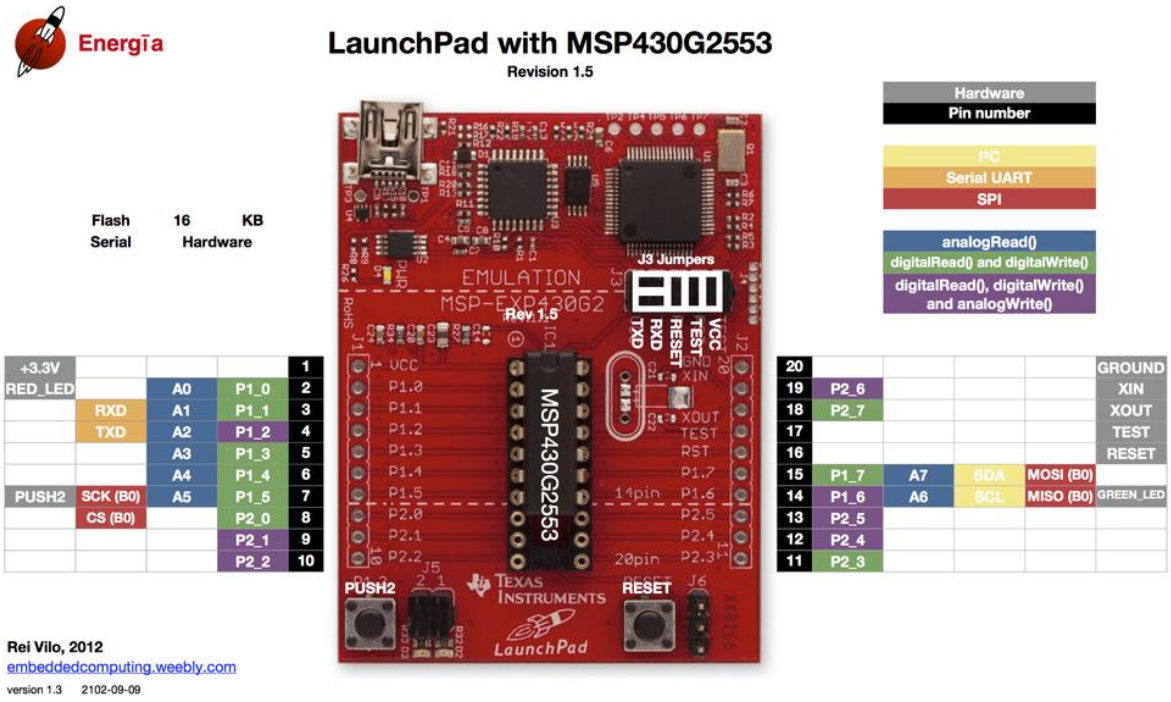
Name/Semester: **Grade:** /20

The MSP430 microcontroller is designed to allow port pins to be individually configured as digital input or digital output and in the case of Port1, all pins P1.0-P1.7 could also be configured as analog inputs. Some pins can be used as analog outputs (analog write). Finally some pins have specific functions, such as serial communication SPI. See the figure below for comprehensive list of all pins functionalities. The labels in gray color are the actual hardware on the Launchpad. Newer versions of the demo board may look a bit different, but essentially the same functionality.



Assemble the circuit below prior to coding making sure to follow diagram. For digital inputs we are using six (6) dip switches connected to port Port2 pins 0-5. If your dip switches have more than 6 positions, ignore the unused ones. Please note the use of external pull-up resisters (1KΩ) with the switches as they are all configured as active-low. When a switch is turned on, it assets a logic zero (GND) on the port pin. For resisters, you can use discrete ones or a resistor network as shown in the photos below.

For digital outputs we are using eight (8) LEDs connected to Port 1 pins 0-7. Please note the use of current limiting resistors (330Ω) with the LEDs on the cathode (GND) side. Also note that LEDs are configured as active high, where a logic high signal (VCC) from the port pin will turn the LED on. Again in the photo below we used a resistor network, but most likely you get discrete resistors in your kit.

To connect the port pins to your breadboard, you need to use Female-Male jumper cables.

If you choose to take MSP430G2553 chip off the Launchpad board and insert it on your breadboard directly, make sure you use a 3.3VDC supply on VCC pin. Also a Reset resistor of 47KΩ must be used to connect Reset pin to VCC.

• **CDA3331C • Intro to Microcomputers •** **Lab Assignment** **4**

Name/Semester: **Grade:** **/20**

[20] 4) Write a program in C that reads a 3-bit desired light pattern from the 3 input switches connected to pins P2.3-P2.5 and displays the pattern on the 8 LEDs (pins P1.0-P1.7) with some "light playing" options based on the input from other 3 switches (pins P2.0-P2.2).

[8] 4.a) Switch on Port2.0 (Read/Rotate mode)

- On (logic 0) – Read mode: Your program continuously reads the switches and takes only the 3-bits (Port 2.3-2.5) as a desired display pattern and displays them on the LEDs (Port 1.3-1.5). All other LEDs on Port 1 are masked to zero (turned off).
- Off (logic 1) – Rotate mode: Once this switch is turned off, your program will rotate the pattern on the 8 LEDs connected to Port1 pins 0-7. The pattern is the 3-bit value you constantly updated and saved during the Read mode.

[6] 4.b) Switch on Port2.1 (Left/Right direction mode)

- On (logic 0) – The LED pattern rotates to the Left.
- Off (logic 1) – The LED pattern rotates to the Right.

[6] 4.c) Switch on Port2.2 (Fast/Slow speed mode) **Hint: Software Delay**

- On (logic 0) – The pattern rotation is Fast.
- Off (logic 1) – The pattern rotation is Slow.

[2] 4.d) **[BONUS]** Modify the program so that the two Switches on Port2.0-1 provide four different fun patterns which continuously display/move on the 8 LEDs. Keep Switch on Port 2.2 to change speed. ***Please be creative***.