

Internet of Things with Intel Edison

GPIO on Edison

Pierre Collet

www.Intel-Software-Academic-Program.com

paul.guermonprez@intel.com

Intel Software

2014-10-14



Credits

Thanks emutex labs !

Emutex documentation Edison GPIO was very useful to write this doc.

Here's the full article :

<http://www.emutexlabs.com/project/215-intel-edison-gpio-pin-multiplexing-guide>

Thanks emutex labs !



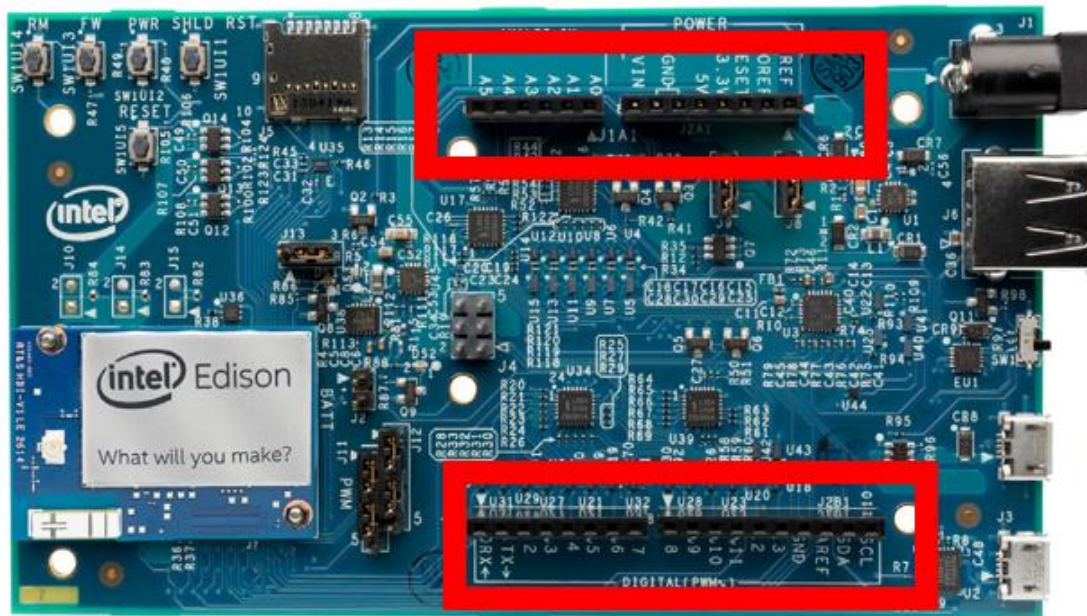
GPIO - Hardware

GPIO ?

Definition

GPIO stands for General Purpose Input Output. GPIO are for instance used to enable communication with the pins on the Edison board.

That's a major difference between your typical PC platform and an IoT platform. You can interact with some sensors, electronics and not only smart USB devices



GPIO – Software Setup



GPIO – Basic setup

Low level access to GPIO

In order to communicate with the pins we have to write some data in some files located at
`/sys/class/gpio.`

Each GPIO you use have to be enabled by **writing** its number in the file:
`/sys/class/gpio/export`

On Linux, everything is a file. And it's easy to write a value in a file.

From now on a new directory is created :
`/sys/class/gpio/gpioXX`

where XX is the GPIO's number you enabled.

GPIO – Basic setup

Low level access to GPIO

Each GPIO has a direction. “in” and “out” are the two possible directions. The direction is set by writing those value in:

`/sys/class/gpio/gpioxx/direction`

From then on we can write or read the value of the GPIO in the file (read if direction=in write if direction = out):

`/sys/class/gpio/gpioxx/value`

Nothing is hidden

Just remember nothing is hidden.

Everything is accessible from regular Linux settings, and with Linux everything is a file. Easy, right ?



GPIO – Advanced Setup



GPIO – Advanced setup

Advanced settings

GPIO basic settings are simple : in or out.

Just write the value in the GPIO configuration file.

But advanced features can be complex. You can define things like the power settings for the GPIO. This kind of setting has to sent to a different GPIO number associated to your GPIO.

Low level : Manual or library ?

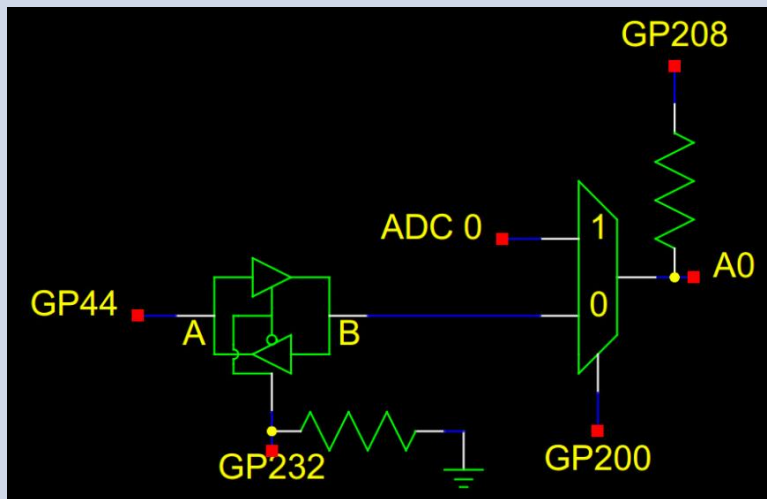
In the following slides, we'll show you detailed diagrams to explain how it works internally because it's important to know you can do it manually if needed.

But in real life, you would use a library and would not think about it.

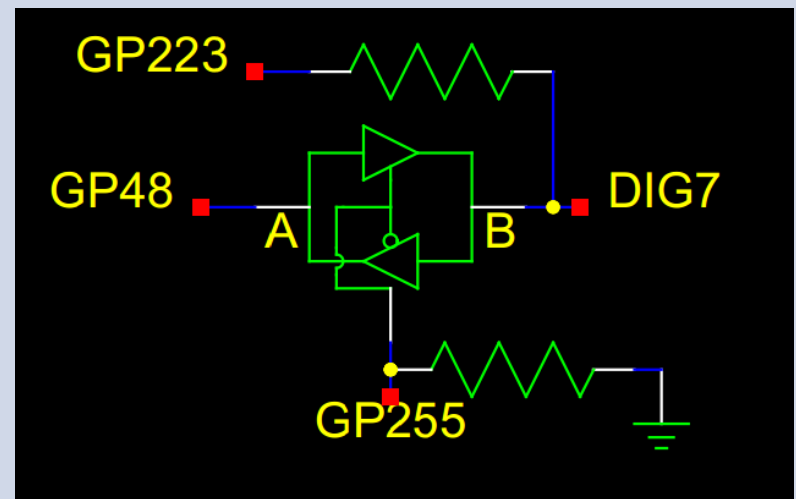


GPIO – Example Diagrams

Analog pin A0



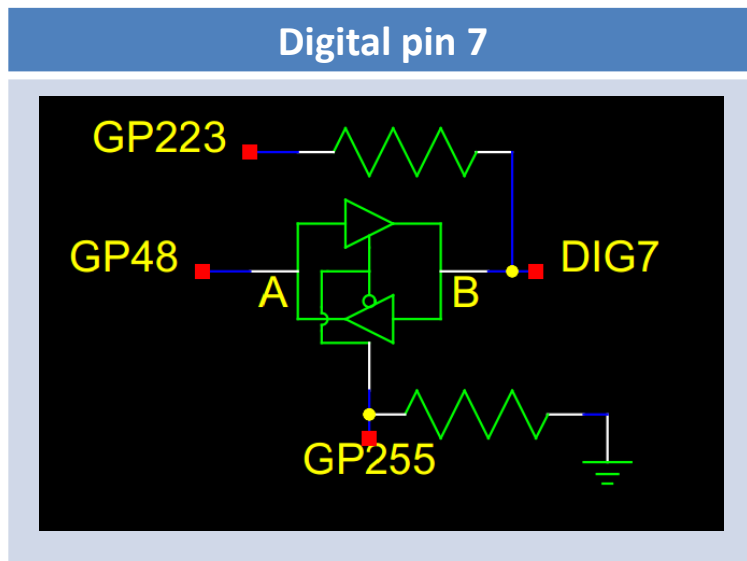
Digital pin 7



GPIO

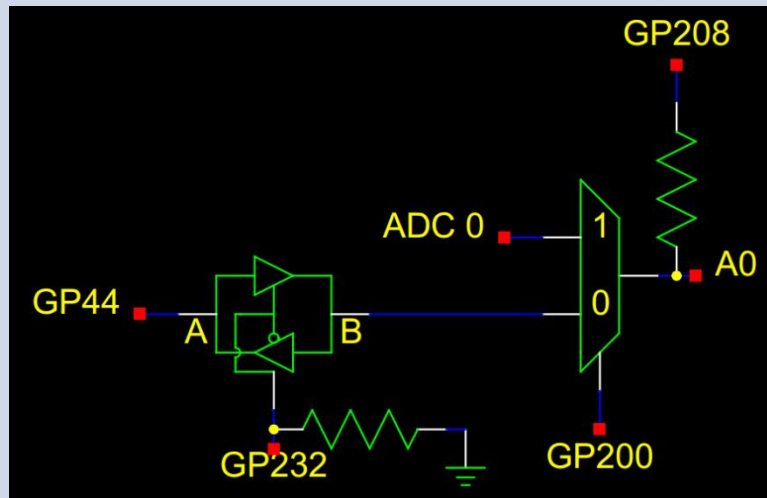
Advanced settings

- 255 : used to set pin 7 as an input or as an output.
If we write “1” in gpio255, we set the pin 7 to as an output and “0” for an input.
- 223 : is a pull-up resistance (47k ohm). It won't going to be used in our examples.
- 48 : It is the one on which we going to write in order to enable or disable the pin 7. If 255 define pin 7 as an input we will read on this GPIO



GPIO

Analog pin A0



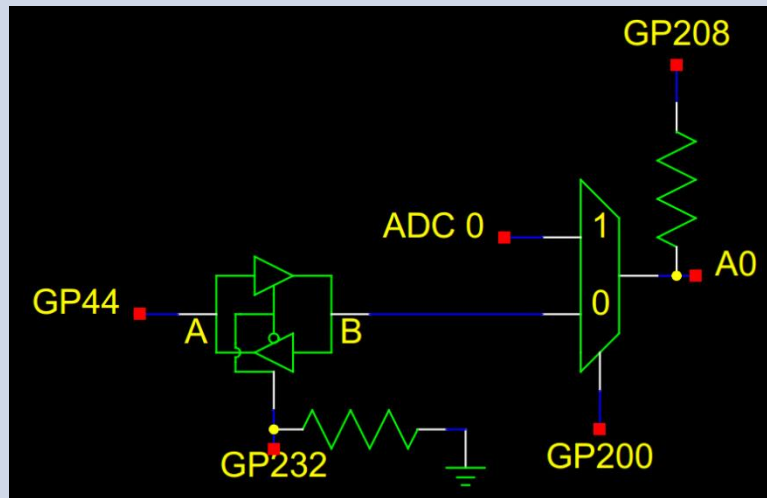
The left part of this diagram is roughly the same as the previous one.

- 200 : Set is a called is like a switch. If “0” is wrote on it, the bottom part of the diagram will be used. “1” for the top part. The meaning of this is that we can use pin A0 as a digital pin (GP200 = 0) or as an analog pin (GP200 = 1)
- 208 is still a pull up resistance as the 223 of the previous diagram.
- ADC0 is used to manage analog signal.

Example: GP200 = 1, GP208 = 1. We can read on **ADC0** the value of a sensor plugged on **A0**. GP208 is set to “1” in order to produce a little electrical signal that can be considered as the 0 value of **ADC0** (when nothing is transmitted on **A0**). This feature is used to reduce the electrical noise on **A0** and so get more accurate values.

GPIO

Analog pin A0



The left part of this diagram is roughly the same as the previous one.

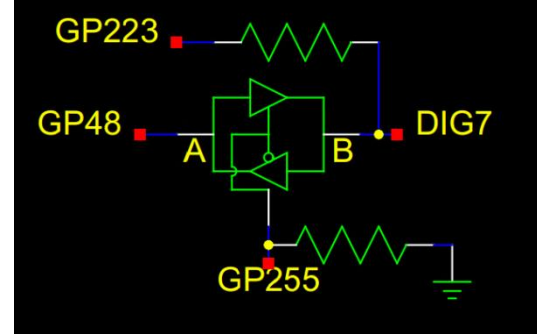
- 200 : Set is a called is like a switch. If “0” is wrote on it, the bottom part of the diagram will be used. “1” for the top part. The meaning of this is that we can use pin A0 as a digital pin (GP200 = 0) or as an analog pin (GP200 = 1)
- 208 is still a pull up resistance as the 223 of the previous diagram.
- ADC0 is used to manage analog signal.

Example: In order to read the analog signal on ADC0 (A0), use this command :

```
cat /sys/bus/iio/devices/iio\:device1/in_voltage0_raw
```

GPIO – Light up a LED

Light up a LED with GPIO



Plugging parts together

Let's plug a LED and a resistor between Digital pin 7 and a ground pin. And check that the jumper on J9 is on the pins 2 and 3.

Enable GPIO

```
echo -n "48" > /sys/class/gpio/export
```

That's the GPIO we need for the pins where we plugged the LED

Enable GPIO for advanced settings

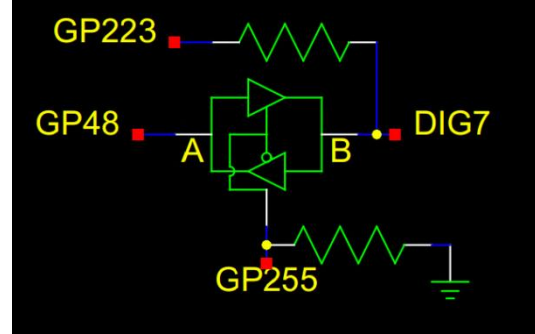
```
echo -n "223" > /sys/class/gpio/export
```

```
echo -n "255" > /sys/class/gpio/export
```

```
echo -n "214" > /sys/class/gpio/export
```

We need them to configure GPIO 48 advanced settings.

Light up a LED with GPIO



Configure directions

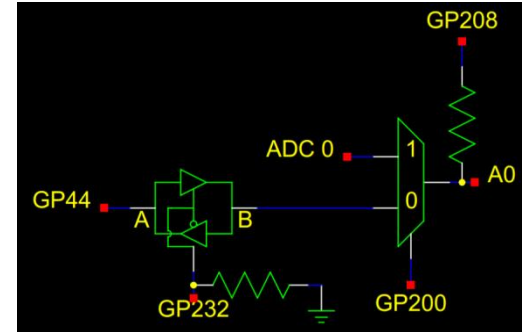
```
echo -n "out" > /sys/class/gpio/gpio48/direction
echo -n "in" > /sys/class/gpio/gpio223/direction # (disable it)
echo -n "out" > /sys/class/gpio/gpio255/direction
echo -n "out" > /sys/class/gpio/gpio214/direction
```

Configure GPIO values

```
echo -n "1" /sys/class/gpio/gpio214/value
echo -n "1" /sys/class/gpio/gpio255/value
echo -n "1" /sys/class/gpio/gpio48/value # (LED is on)
echo -n "0" /sys/class/gpio/gpio48/value # (LED is off)
```


GPIO – Read Sensor Value

Read A0 with GPIO



Plugging parts together

Let's plug a sensor between the pin A0 and the 5V pin.

Enable GPIOs

```
echo "214" > /sys/class/gpio/export  
echo "200" > /sys/class/gpio/export  
echo "232" > /sys/class/gpio/export  
echo "208" > /sys/class/gpio/export  
echo "261" > /sys/class/gpio/export
```

Same concept as previous example, with different GPIO numbers.

Read A0 with GPIO

Configure directions

```
echo "low" > /sys/class/gpio/gpio214/direction
echo "high" > /sys/class/gpio/gpio200/direction
echo "low" > /sys/class/gpio/gpio232/direction
echo "in" > /sys/class/gpio/gpio208/direction
echo "high" > /sys/class/gpio/gpio261/direction
```

Configure GPIO values

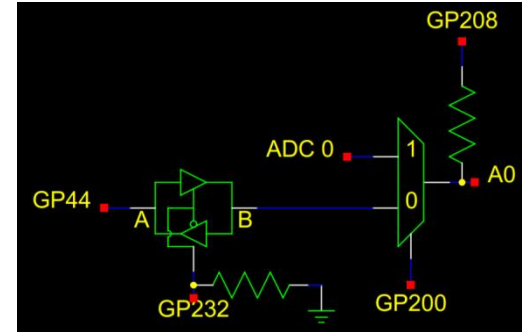
```
echo "1" > /sys/class/gpio/gpio261/value
```

At the end do

```
echo "high" > /sys/class/gpio/gpio214/direction
```

Now you can read the value of A0

```
cat /sys/bus/iio/devices/iio\:device1/in_voltage0_raw
```



GPIO – LedSensor Script

LedSensor Script

LedSensor

In the previous slides we've seen everything can be done manually. We can light up a LED, we can read value from a sensor.

Let's go further and use a script to read value from a sensor, light up a LED if the value is higher than a threshold.

Hardware

Let's plug a sensor between the pin A0 and the 5V pin.

Let's plug a LED and a resistor between Digital pin 7 and a ground pin



GPIO

Script

You can find a shell script at this path: `labs/ledsensor/bash/ledsensor.sh`

Run

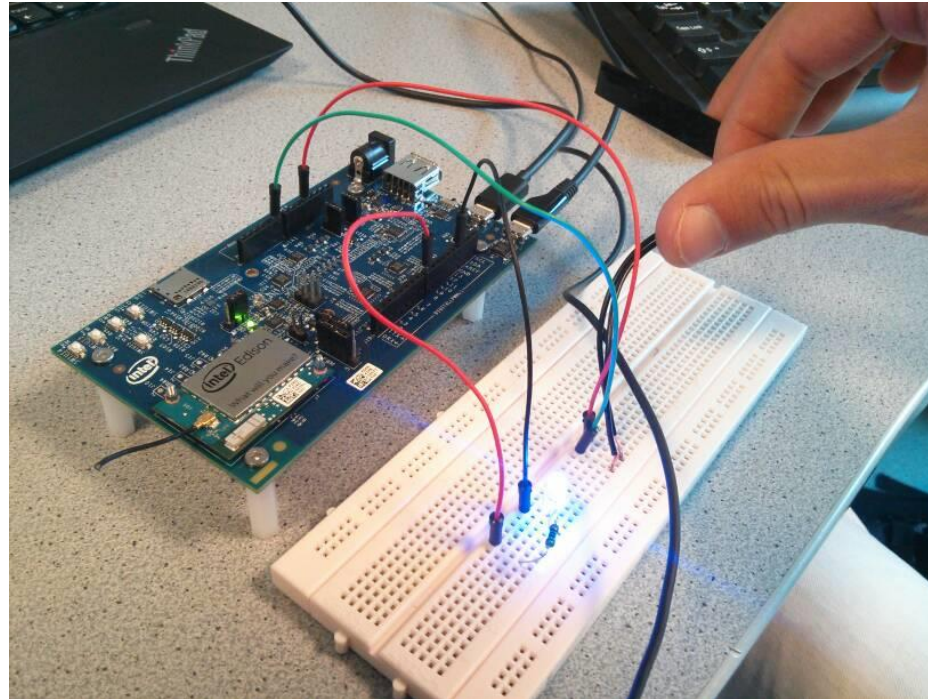
Simply enter the command : `sh ledsensor.sh`

Now you see the values of the sensor on your screen.

Press it or bend it (depending on what kind of sensor it is)

When you press or bind it the LED light light up!

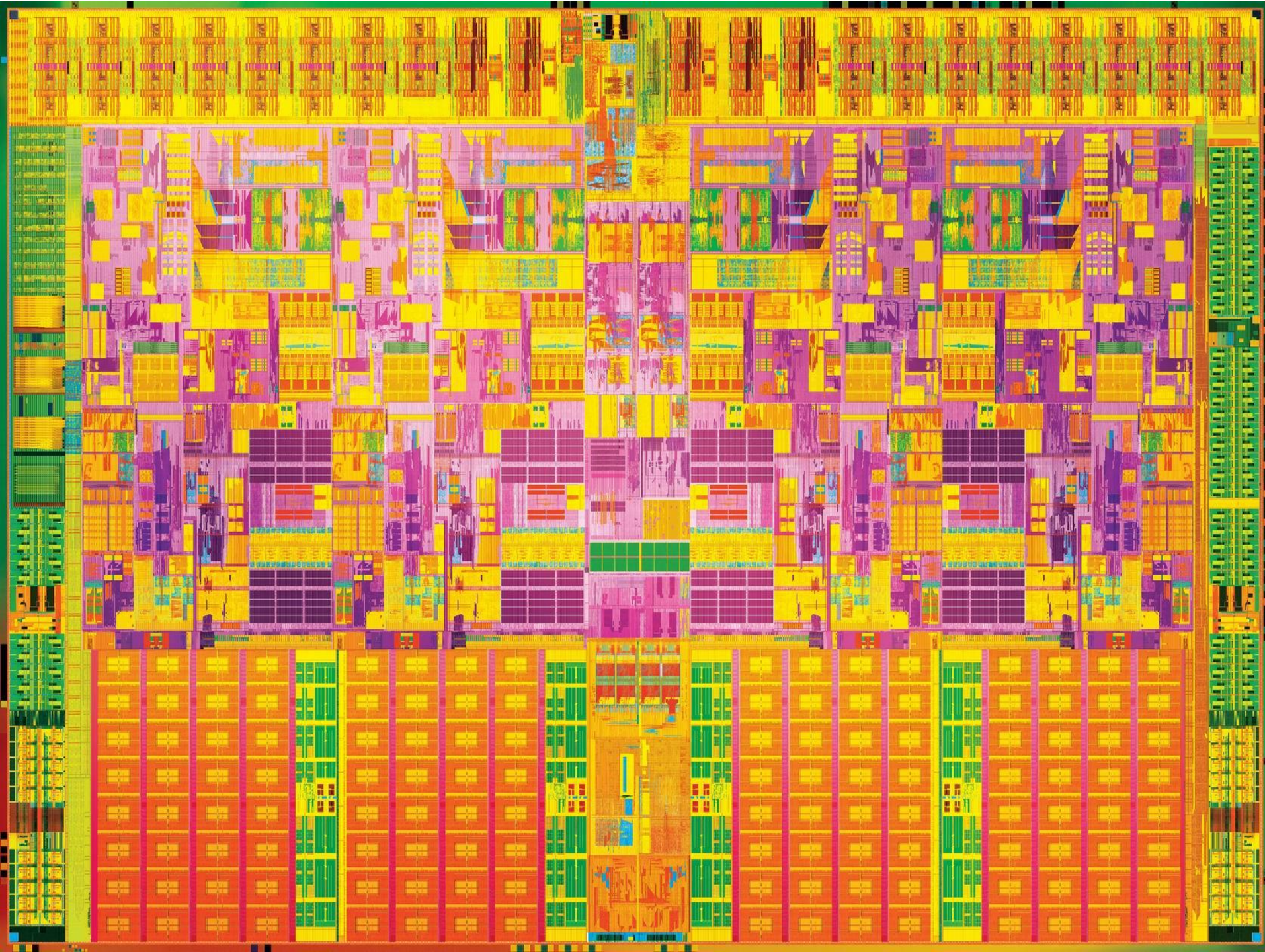
Setup



The diagram shows a 1-bit DAC circuit. An operational amplifier (op-amp) is configured with its non-inverting input (+) connected to GP44 and its inverting input (-) connected to GP232. The op-amp's output is connected to the 'B' input of a 2-to-1 multiplexer. The multiplexer's 'A' input is connected to GP44. The multiplexer's select line is connected to ADC 0. The multiplexer has two data outputs: output '1' is connected to GP200, and output '0' is connected to GP208. A resistor is connected between GP208 and A0, and another resistor is connected between GP200 and A0. The output of the DAC is A0.

However, it also shows you that even if you can do everything, you better use a library to simply use you board.
That's the topic of the next course (libmraa).





License Creative Commons – By 3.0

You are free:

- **to Share** — to copy, distribute and transmit the work
- **to Remix** — to adapt the work
- to make commercial use of the work

Under the following conditions:

- **Attribution** — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

With the understanding that:

- **Waiver** — Any of the above conditions can be waived if you get permission from the copyright holder.
- **Public Domain** — Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.
- **Other Rights** — In no way are any of the following rights affected by the license:
 - Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;
 - The author's moral rights;
 - Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.
- **Notice** — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.

<http://creativecommons.org/licenses/by/3.0/>