Intel Edison

Intel(R) Edison is a dual-core Silvermont Atom(TM) clocked at 500MHz. The Edison also features 4GB of storage, 1GB ram and on-board WiFi and Bluetooth.

Currently supported boards:

- Intel Arduino board
- Intel breakout board

UART

On both the Arduino board and the breakout board, The available UART interface is on /dev/ttyMFD1

Intel Arduino board

The Edison used with the Arduino board has the following limitations in libmraa:

- Do not use the 'reset' button on the arduino shields, there is a hardware bug and the platform will likely not come back up correctly
- I2C is exposed on i2c-6, therefore you must use bus 6 and not bus 0
- PWM available on default swizzler positions. (3,5,6,9)
- SPI exposed is also used for the ADC. Try not to use your own CS.
- Max SPI speed is 25Mhz/4 ~6.25Mhz
- SPI PM can sometimes do weird things you can disable it with: `echo on > /sys/devices/pci0000\:00/0000\:00\:07.1/power/control`
- ADC kernel module will return 16bit number but the ADC itself only has an accuracy of maximum 12bits and in MRAA it's limited to 10bits by default. Use `mraa_aio_set_bit(12)` to switch to the maximum resolution mode. This ADC is only included on the Arduino board.
- AIO pins are treated as 0-5 in **`mraa_aio_init()`** but as 14-19 for everything else. Therefore use `mraa_gpio_init(14)` to use A0 as a GPIO
- Arduino pin 7 can sometimes negatively impact the WiFi capability, if using WiFi avoid using this pin
- Edison's i2c-1 can be used using for example the sparkfun i2c breakout ontop of the Arduino breakout board, this is not supported officially so asking for mraa_i2c_init(1) will result in getting i2c bus 6 (the default one). However using raw mode (mraa_i2c_init_raw(1)) this bus is fully usable
- If you want to use /dev/ttyMFD2 you have to use the raw uart mode passing a std::string or char* argument of "/dev/ttyMFD2" to mraa:Uart() or mraa_uart_init_raw. By default there is a getty running on that interface so you will need to disable that first

Because of the way IO is setup with the tristate on the Arduino breakout board IO will be flipped as it is setup. It's recommended to setup IO pins & direction before using them in a `setup()` method or similar. It's impossible on this platform to avoid some GPIOs flipping on setup.

Intel(R) breakout board