Intel Edison Cheatsheet

A comprehensive cheat sheet covering Intel Edison, its features, setup, and common commands. Useful for developers and makers working with this platform.

Edison Basics & Setup

Edison Specifications

CPU:	Intel Atom System-on-Chip (SoC), 500MHz
RAM:	1 GB LPDDR3
Storage:	4 GB eMMC
Wireless:	Wi-Fi 802.11a/b/g/n, Bluetooth 4.0
USB:	1x USB 2.0 OTG
GPIO:	40-pin connector with GPIOs, UART, I2C, SPI

Setting up Edison

1. **Connect Edison to your computer:** Using a USB cable.
2. **Install drivers:** For Windows, drivers might be needed.
3. **Use a serial terminal:** Like PuTTY or screen to connect via serial communication.

Default serial settings: `115200` baud rate, `8` data bits, `no` parity, `1` stop bit.

Initial Login

Username:	root
Password:	(None by default, set it using `passwd`)
First Steps:	Set a password and configure Wi-Fi.

Networking & Package Management

Connecting to Wi-Fi

Use the `configure_edison --wifi` command to scan for and connect to a Wi-Fi network.

Alternatively, manually configure Wi-Fi by editing `/etc/wpa_supplicant.conf`.

Example:

```
network={
 ssid="YourNetworkName"
 psk="YourWiFiPassword"
 key_mgmt=WPA-PSK
}
```

Networking Commands

`ifconfig wlan0`	Display Wi-Fi interface configuration.
`iwconfig wlan0`	Display wireless network configuration.
`ping <address>`	Test network connectivity.
`dhclient wlan0`	Obtain IP address via DHCP.

Package Management (opkg)

Edison uses `opkg` for package management, similar to `apt` or `yum`.

- `opkg update` - Update the package lists.
- `opkg install <package>` - Install a package.
- `opkg remove <package>` - Remove a package.
- `opkg list` - List available packages.
- `opkg upgrade` - Upgrade installed packages.

Working with GPIOs

Accessing GPIOs

GPIO pins can be accessed via the command line using the `gpio` command. Libraries are also available for Python and other languages.

GPIO Commands

`gpio help`	Display help information.
`gpio export <pin> <direction>`	Export a GPIO pin for use (direction: `in` or `out`).
`gpio unexport <pin>`	Unexport a GPIO pin.
`gpio read <pin>`	Read the value of a GPIO pin.
`gpio write <pin> <value>`	Write a value (`0` or `1`) to a GPIO pin.

Example: Blinking an LED

Connect an LED to GPIO pin 13 (for example) with a suitable resistor.

```
gpio export 13 out
while true; do
 gpio write 13 1
 sleep 1
 gpio write 13 0
 sleep 1
done
```

Development & Programming

Programming Languages

Edison supports multiple programming languages including C/C++, Python, Node.js, and others.

Python Development

Libraries:	`mraa` (for GPIO access), `pyupm` (for sensors).
Install Libraries:	`opkg install python-mraa python-pyupm`

Example (GPIO):

```
import mraa
import time

led = mraa.Gpio(13)
led.dir(mraa.DIR_OUT)

while True:
 led.write(1)
 time.sleep(1)
 led.write(0)
 time.sleep(1)
```

Node.js Development

Libraries:	`mraa` (for GPIO access), `upm` (for sensors).
Install Libraries:	`npm install mraa upm`

Example (GPIO):

```
var mraa = require('mraa');
var led = new mraa.Gpio(13);
led.dir(mraa.DIR_OUT);

setInterval(function() {
 led.write(led.read() ^ 1);
}, 1000);
```