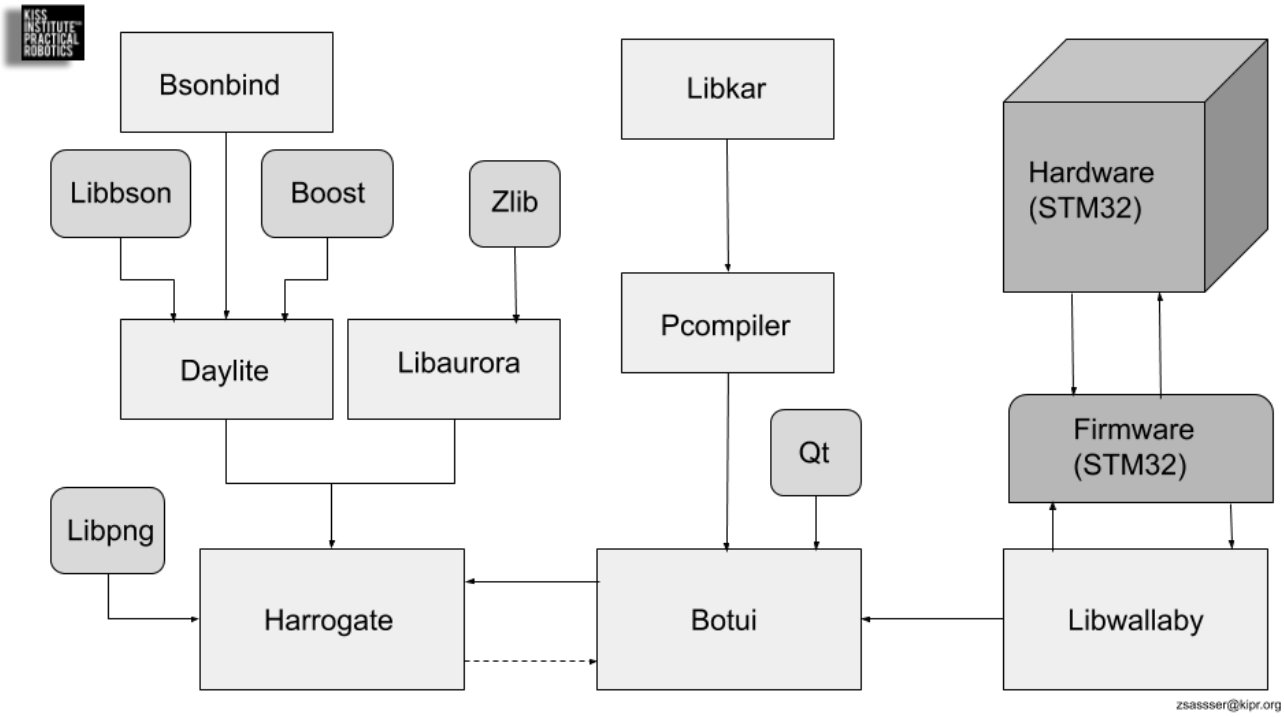
KISS INSTITUTE FOR PRACTICAL ROBOTICS

Wombat Developer Manual

Author(s): Zachary Sasser

Overview of KIPR-Suite



Program	Description
Libwallaby	Library for interfacing with the controller hardware.
Botui	UI for Interfacing with the controller
Harrogate	Node.js webserver for KISS-IDE
Pcompiler	Used to compile KISS-IDE projects
Libkar	Archiving tool used by Pcompiler
Daylite	Networking Backbone for Harrogate
Libaurora	Graphics Library that interfaces with Daylite
Bsonbind	Utility that generates C++ structs for descriptions of BSON

Note: These program descriptions are over-simplified, visit their respective github pages/documentation for more information

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Wombat Developer Manual

Operating System Installation:

Linux/MacOS/Unix:

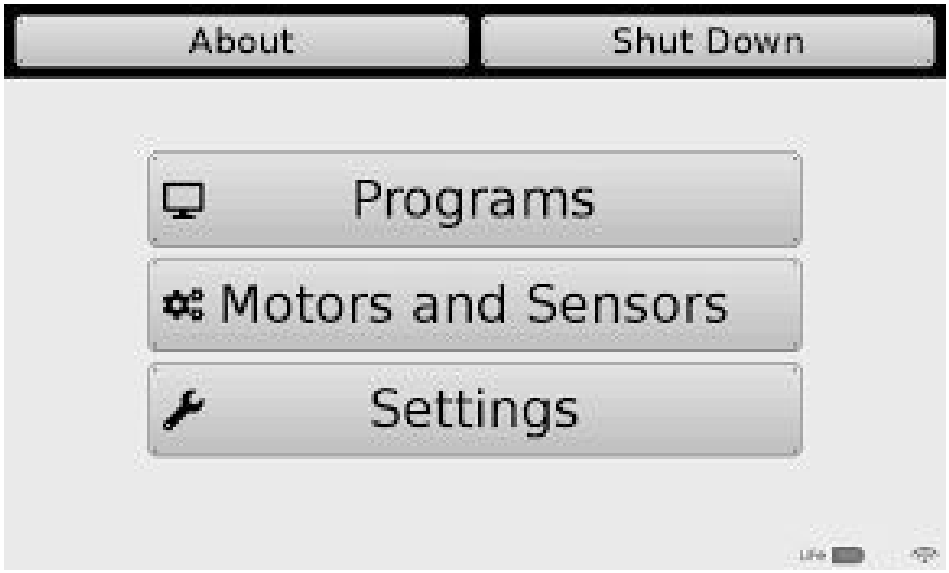
1. Download the latest KIPR-OS image: Click Here (Dropbox)
2. Most Linux Distros come with dd installed, but if not run the following:
    a. “sudo apt-get install gdebi”
3. Insert an SD card and find which pseudo-directory it is.
    a. “sudo fdisk -l“
    *b. Alternate: “*sudo df”
4. If your drive is /dev/sdb (/dev/sdb1 is a partition of sdb), then you would type
    a. “sudo dd if=”Wombat.img” of=/dev/sdb bs=4M status=progress

Windows (or Linux/MacOS):

1. Download Balena Etcher and install it to your PC: https://www.balena.io/etcher/
2. Download the latest KIPR-OS image: Click Here (Dropbox)
3. Open Balena Etcher and follow the simple prompts, choose your Wombat image and your SD card

On Startup:

The first time you boot up your pi, it will run fsck which is similar to CHKDSK in MS-DOS. Then it will reboot and boot into this screen; this is Botui. To get to the desktop, go to “*Settings* *Hide UI”* or hit “*WIN+D”*.



KISS INSTITUTE FOR PRACTICAL ROBOTICS

Wombat Developer Manual

Flashing the STM32:

When the Wombat is first assembled or the STM32 gets out of sync, you have to write the firmware binary to the STM32 processor (the device that handles the hardware). To flash the processor, navigate to /home/pi and then type “sudo ./wallaby_flash”, The output should look like this:

```
Wombat:~ $ sudo ./wallaby_flash
/sys/class/gpio/gpio17 - initializing gpio pins
echo 1 > /sys/class/gpio/gpio17/value
resetting co processor...
stm32flash -v -S 0x08000000 -w /home/pi/wallaby_v8.bin /dev/ttyAMA0
stm32flash 0.4

http://stm32flash.googlecode.com/

Using Parser : Raw BINARY
Interface serial_posix: 57600 8E1
Version      : 0x31
Option 1     : 0x00
Option 2     : 0x00
Device ID    : 0x0419 (STM32F427/37)
- RAM        : 192KiB  (8192b reserved by bootloader)
- Flash      : 1024KiB (sector size: 4x16384)
- Option RAM : 16b
- System RAM : 29KiB
Write to memory
Erasing memory
Wrote and verified address 0x080030f4 (100.00%) Done.

echo 0 > /sys/class/gpio/gpio17/value
resetting co processor...
```

If you are getting the “/sys/class/gpio/export: Permission denied” error:

- Try “sudo chmod +x wallaby_*” or “sudo chmod 777 wallaby_flash”
    - (Security doesn’t really matter for this)
- You are in the wrong terminal (use root not tty#)
- The GPIO cannot connect to the STM32 (usually because of a race condition)
- There is something blocking the output of the GPIO pins

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Wombat Developer Manual

Hardware Overview

Work in Progress...