Getting started

Once you have received your kit you should take some time to familiarise yourself with the basics of the equipment you will transform into an intelligent and autonomous robot. Start small, resist the temptation to chain every component together just to see what happens. Your first step should be to do something simple like connect the power-board to the module, then attach the power cable or a battery and try to login to your module.

Robot building essentials

Following links provide some basic tips and tricks when building robots using lego:

- Mechanical design (gears, bracing, ...): http://www.cs.tufts.edu/comp/150IR/artoflego.pdf
- Gear trains and bracing: http://www.clear.rice.edu/elec201/Book/legos.html
- Gear trains and wheels: http://www.ecst.csuchico.edu/%7Ejuliano/csci224/Slides/03%20-%20Gears%20Pulleys%20Wheels%20Tires.pdf
- Motor characteristics: http://www.philohome.com/motors/motorcomp.htm
- Tips and tricks: http://homepages.inf.ed.ac.uk/gde/work/sdp/Robot_Construction_Guide.odt
- Gear ratio calculator: http://gears.sariel.pl

Robot programming

The code (including the Sandbox itself) is available from bitbucket.

Documentation and example snippets are in the wiki.

Developing on DICE or on your own machine

To develop the code for this practical you will need a Linux computer with the following dependencies installed:

- Python 2.7 - with numpy and ideally ipython and/or and IDE (for development)
- OpenCV 3.0.0 - compiled with python and GUI support (e.g. GTK)
- Phidgets library - compiled c++ library with Python wrappers available from the same website

All of these required for running the code, even if you don't intend to use camera or Phidgets devices on your development machine. The DICE machines in the lab have these set up.

To develop your code, download the Sandbox into a directory on your machine. You can also use git: **git clone git@bitbucket.org:IPAB-SLMC/rss-sandbox.git**

The user code goes inside the **toddler.py** file.

You can run the code by executing: **python sandbox.py**

For documentation and example snippets have a look at the wiki.

Connecting the Module

When the module is powered it will boot-up automatically. The full boot-up sequence takes around 25 seconds, so remember to wait for a short interval before attempting to login to the module. You can connect to the module using an Ethernet cable with one end plugged into an Ethernet port in the robotics teaching lab, and the other connected to the Ethernet port on your module. Once a cable is connected your module will detect it automatically and identify itself to the DICE network. The module will also report its status to a server. You can check the status of your FitPC on this address: http://handy.inf.ed.ac.uk/tc/list.php (accessible only from within the university network). The status will update every minute when the robot is connected to the network.

Logging In/Out

Now you're ready to login to your module. Bring up a terminal window on a DICE machine and use the **ssh** command to connect to the module. The name of your module will be printed on its top side, for example the module on my desk is named Panda. So to login to Panda as student I would type **ssh -XC student@Panda**. If you get a connection error (ssh: connect to host Panda port 22: No route to host) try replacing the name of the FitPC with its IP. You can find the IP of each fitPC at http://handy.inf.ed.ac.uk/tc/list.php . (This assumes that the status of your FitPC is OK).

Remember to add the **-X**C option so that you can later start graphical text editor and debug your image processing remotely. After typing this command you will be asked for a password. The password is **password**. If all that went well you are now able to login to the module. You should store all your programs in **/home/student/** directory. To log out type **exit**. If you want to reboot or turn off the FitPC use **sudo reboot** and **sudo poweroff**. Always turn off the FitPC using **sudo poweroff**, otherwise you may cause damage to the hard drive and loose your work.

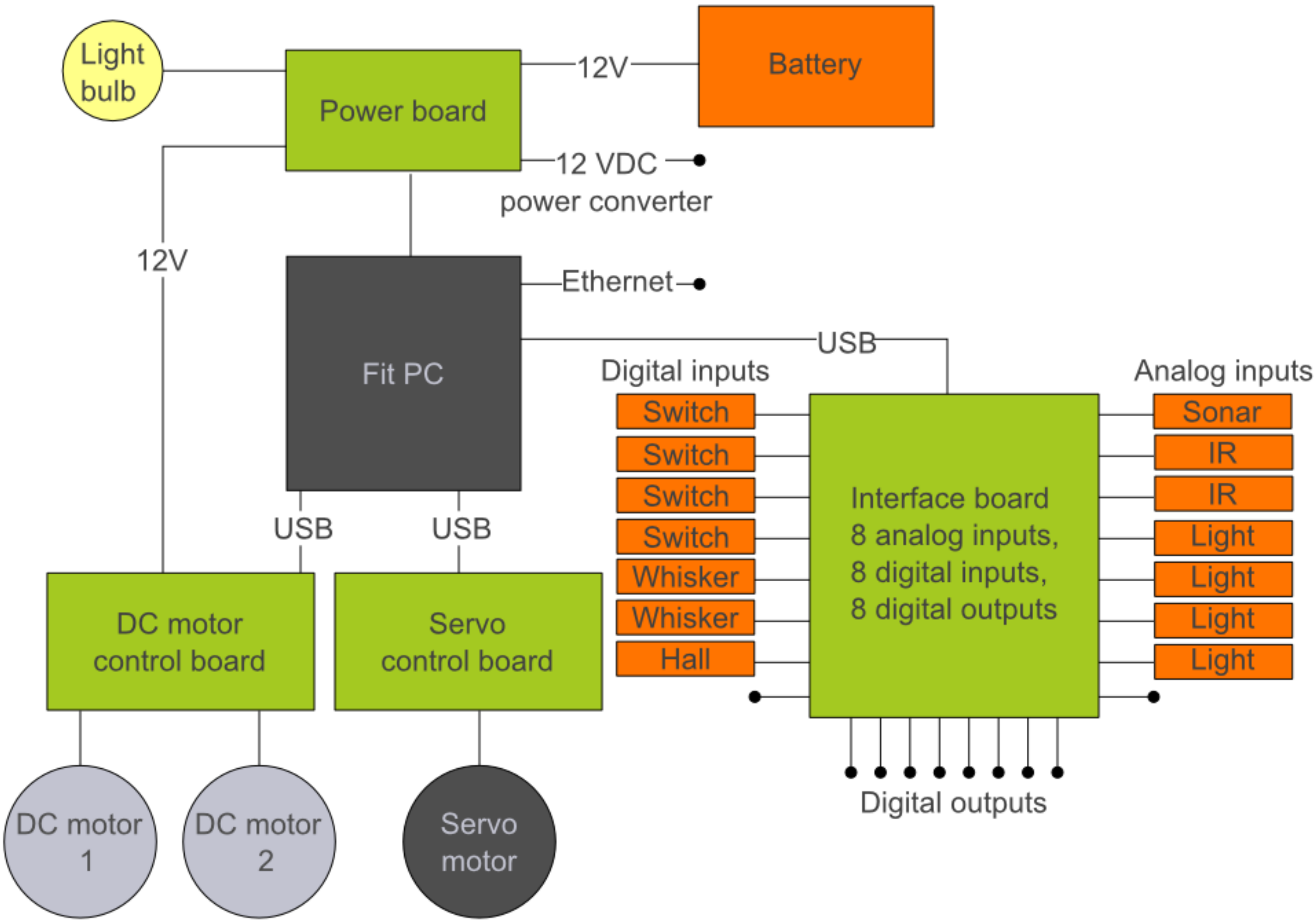
File Transfer

See this section in the wiki.

Step by Step

1. Login to the brain brick as **student** using password **password** (example for brain brick named Panda): **ssh -XC student@Panda**
2. Write your code on the brain brick or copy it from the DICE machine
   1. You can use following text editors on the brain brick: vim (**vi**), nano (**nano**) or graphical editor (**editor**)
   2. If you prefer to write your code on the DICE machine you can copy the code over using FTP.
3. Run program by executing the **sandbox** command on the FitPC (good for debugging) or by pressing the power button twice.

System diagram


Light bulb
Power board
12V
Battery
12 VDC
power converter
12V
Fit PC
Ethernet
USB
Digital inputs
Analog inputs
Switch
Switch
Switch
Switch
Whisker
Whisker
Hall
Interface board
8 analog inputs,
8 digital inputs,
8 digital outputs
Sonar
IR
IR
Light
Light
Light
Light
USB
USB
DC motor control board
Servo control board
Digital outputs
DC motor 1
DC motor 2
Servo motor