

Frequently Asked Questions

- **How do I know that the FitPC has started properly and is ready?**
Check the status at <http://handy.inf.ed.ac.uk/tc/list.php>
- **Can I work with the kit outside the lab?**
No. Also keep your kit and Lego in the locker when you are not using it.
- **Can I leave the batteries at the charger?**
Yes. You can even exchange batteries between groups, just make sure you can return two batteries when the course ends.
- **I cannot SSH to the FitPC, what is the problem? (No route to host)**
Make sure you have plugged the ethernet cable and power cable properly.
Plug the cable to one of the wall ports, not the DICE machine itself.
If you cannot login, don't change ports (on the wall). The network remembers the port and it might take several minutes until it gets reassigned.
Try turning the brain brick **on and off** using the power button (not by removing the power).
- **I cannot get into the lab, my swipe card is not working.**
After 5pm and over the weekend you have to enter your pin as well.
If this happens during the first week, you may not have the permissions to access the lab yet.
Just wait until it gets sorted out. It usually takes several days.
If the problem prevails, contact the demonstrators.
- **My hardware is not working, what should I do?**
First try to recreate the problem using original toddler program to make sure that the problem is not within your code. The original toddler.py is always stored on the FitPC at `/home/tc/toddler.py`. If you remove/rename the toddler.py in your home directory, the original toddler.py will be used.
Check the wires.
If you are certain that your piece of hardware is faulty, place it into box labelled "Faulty IAR kits" located on top of the locker cabinet and note down your locker number and nature of the problem on provided form.
- **Can we use our own padlocks?**
NO!!!
- **Can we remove the straps or lego parts from the brain brick or one of the sensor boards?**
NO!!!
- **Can I install 3rd party software on the FitPC?**
NO
You don't have enough user rights to do that. Stick to available software.
- **How do I start my program on the FitPC?**
Press the power button twice.
It will start the Sandbox, as long as it's not running already.
- **How do I stop/restart my program on the FitPC?**
Press the power button twice, it will kill any running Sandbox programs.
- **My program stops when I end the SSH session using "exit" command.**
When you start the sandbox via "python sandbox.py" or via "sandbox" command on the FitPC, the session will terminate when you disconnect the FitPC.
To start your program remotely, press the power button on the FitPC twice.
- **How can I turn off the FitPC safely?**
sudo poweroff
sudo reboot
- **What are the LEDs trying to tell me?**
When the Sandbox is not running, the LEDs are off.
When the Sandbox starts, the all the LEDs will flash 3 times.
When the Sandbox runs normally, the green LED will flash like a slow heartbeat and the yellow

and red one will be under your control.

When the Sandbox stops due to an unhandled error, the green LED will flash fast.

- **Where did the output of "print" go?**

When running the sandbox on the FitPC (started using the power button), the standard output and the error output streams get dumped into a log file: /tmp/sandbox/log.txt

You can SSH into your FitPC and copy this file over, or you can create a symbolic link in your home directory using: `ln -s /tmp/sandbox/log.txt /home/student/`

You can then copy the log file via FTP.

- **What happens when my code breaks the Sandbox?**

All errors, including syntax, typos, and system calls will be handled quietly and recorded in the log file. When the error occurs, the green LED will start flashing fast.

- **Can I use other programming languages like c/c++, Java, c#, Fortran, Lisp, Assembler...? NO!!!**

You have to use Python for this practical.

- **But c++ is the best thing that happened to humanity since we learned how to use fire, why can't I use c++?**

c++ introduces a lot of easy-to-cause technical problems, very few people (even at the masters level) have proper training to use it correctly, and having to compile the code slows down the development greatly. The run-time speed-up is negligible within the scope of this practical and the low level implementation of OpenCV and Phidgets is running optimized c++ libraries anyway. The Python interface adds only minimal overhead but greatly improves the productivity and decreases frustration with obscure errors.