# PIR guidance notes – how to set up and use

## Notes on how to use the PIR setup

### Equipment:

Up to 6 PIR Sensors + 1 LDR 🡪 1x Arduino 🡪 Raspberry Pi

LDR=light dependent resistor

### Sensor

The sensors consists of an infrared panel under the removable cap. This is a lens that has a wide viewing angle to visualise the entire cage. There are four beams that overlap and movement is detected when these beams are broken. The cap acts to focus these beams. Binary coding for movement where; 1=movement, 0=no movement.

The casing of the sensor has been 3D printed with additional adaptations 3D printed for the custom made mouse cages for Vlad’s lab. The 3D printer uses pre-made designs which can be ordered/downloaded online.

Note: it would be good to update the design to have longer sides so that the sensors are less likely to fall.

### Telephone cables

These are used to connect the sensors to the Arduino. To make cables; first strip of shielding to expose wires. The exposed wires can then be slotted into the plastic clip/connector. The release clip should be aligned to the line/indent on the cable shielding.

### Arduino

Receives inputs form sensors via telephone cables and outputs signals via a USB cable to computer/raspberry pi. Arduino sums the inputs over 10 seconds and sends signals to raspberry pi/computer.

Connected as follows:

* Power 5V and ground.
* A0, A1, A2 = LDR.
* 1-6 = PIR sensors.
  + Note each sensor goes to 1 pin with a common ground.

### Setting up PIR system

* Plug in Arduino to raspberry pi/computer
* Tools 🡪 port 🡪 USB port
* Tools 🡪 serial monitor
  + This opens up a window that prints a live readout of the data every 10 seconds (may differ depending on settings defined in script).
* Arduino reads PIR sensors using a java script code (Arduino software can be downloaded below).
* Set the pins associated with each PIR sensor in the code.
* Set delay and loop counter – see comments in script for where to edit).
  + Commonly used; loop counter = 100, delay = 99 (reads every 1 second and sends data every 10 seconds).
* Once happy with the script this can be ‘uploaded’ to the Arduino using the buttons above the script.
* Example 2 second delay, reads every 10 seconds:
  + 1. Two second delay
  + 2. Reads state of each PIR – if movement detected (1) add 1 to the total movement.
  + 3. After 10 seconds prints off arduino ID, a total movement value per PIR (up to 6 PIRS) and a 7th value for the LDR.
    - VVEEG\_1 13 12 15 17 -100 -100 603
      * Note: the 5th and 6th PIRs can be arbitrarily set to any number if not in use; in this example they were set to -15.
      * Can also comment out 4 lines in the script to just not print values for the 5th and 6th PIRs.
* First it is important to confirm that the sensors work and which sensor corresponds to which PIR on the system
  + Troubleshooting tip: change the settings of the loop counter to speed up how often data are averaged and printed to the monitor.
* If the monitor just prints symbols then something is wrong. This is often due to the read speed of the monitor window (bottom right corner) being set to a value different to that stated in the script. Script = serial.begin(57600); monitor window = 57600 baud.

### Saving data

* Connect Arduino to raspberry pi/computer using micro USB to UBS connector.
* When setting up a new raspberry pi.
  + Download raspberry pi noobs.
  + Copy onto SD drive.
  + When you turn on the pi it runs through the installation.
  + It may be necessary to set the time of the raspberry pi
    - Terminal 🡪 sude date -s “Thu Aug 9 21:31:26 UTC 2012”.
* Activating server to read Arduinos.
  + Click the button with the symbol raspberry (top left) 🡪 programming 🡪 node red OR IF NOT USING RASPERRY PI open the terminal 🡪 “node-red”
  + Open internet browser and copy the IP address stated in node red into the web address bar.
  + This opens up the server and shows circuit diagram of how files are named and saved.
    - Flow chart/circuit diagrams can be imported from the clipboard or saved files.
* NOTE: node red can be setup to automatically open when the pi is turned on (especially important in cases of automatic restarts to prevent data loss).
  + Open node red and on screen should be instructions as to how to do automatic restarts – copy this into the terminal and run.
* This will automatically start receiving data from the arduinos as soon as they are plugged into the pi.
  + Every time it receives a signal from the Arduino it adds a timestamp to the value and appends to data
* Files save as .csv

### SOFTWARE DOWNLOADS

Download this for running the arduino on your computer.

<https://www.arduino.cc/en/Main/Software>

(software 🡪 downloads 🡪 Arduino 1.8.9)

In order to save PIR data download node-red

<https://nodejs.org/en/download/>

Once downloaded follow instructions in to install node red:

<https://nodered.org/docs/platforms/windows>

Once node red installed open terminal and type ‘node-red’. This will start the server and the IP address generated can then be used in a browser to access flow diagrams about saving.