OpenCv tries to get cam feed

Gets cam feed? — No → Warning MSG

Yes

640x480

Reads images as frames → Writes frame to disk as greyscale

cv2.imwrite(..., frame[:, :, 1])

At 30Hz ~120K images:
3600K per sec -> 3.6MB per sec
311040 MB per 24hr
311 GB per 24hr

HDD

Data Analysis OFFLINE

Cronjob save to drive every x minutes

https://answers.opencv.org/question/75649/object-detection-and-splitting-clustering/

Split each frame into n frames (n = number of cages) ← Yes — Multiple cages?

No

1. Binarize (mice vs reference)
2. Noise removal (?)
3. Calculate moments of binary image
4. Get centroid
5. Between n-points calculate distance traveled by centroid
6. Calculate acceleration
7. Others

Cronjob:
- Get batch img 1-100, 101-200, ...

csv -> x, y, ...

```
    .
elif command == "
    usb.write(b'le
    print("Arduino
elif command == "
    usb.write(b'le
    print("Arduino
elif command == "
    print("Exiting
```

Controlling arduino via USB

```
l":  # turn on Arduino LED
d_on')  # send command to Arduino
 LED turned on.")
k":  # turn off Arduino LED
d_off')  # send command to Arduino
 LED turned off.")
x":  # exit program
 program ")
```

```
print('Exiting
exit()
```

```
try:
    usb = serial
```
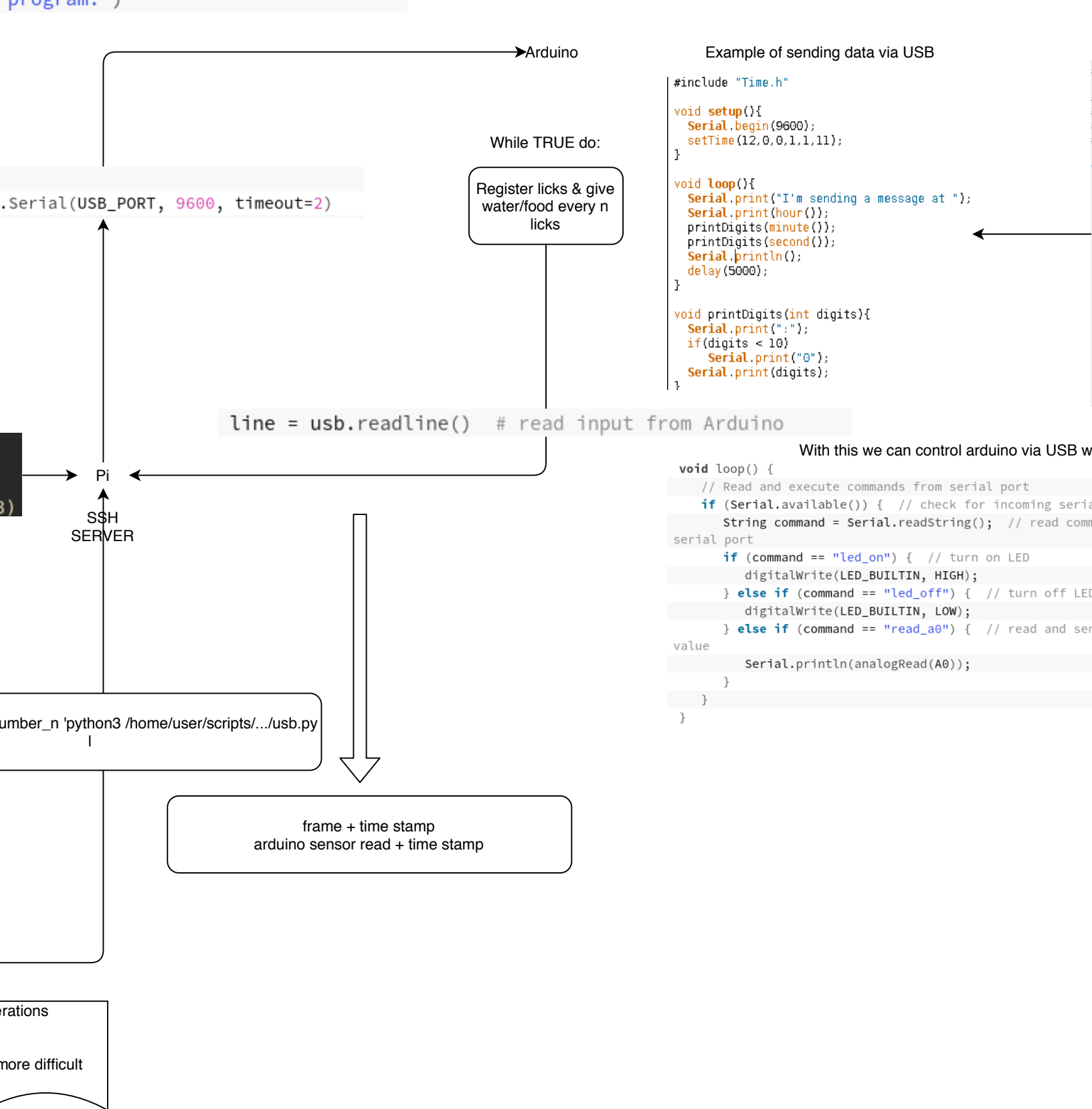
To each frame read put this timestamp

```
>>> from datetime import datetime
>>> from time import mktime
>>> datetime.now()
datetime.datetime(2020, 8, 10, 11, 39, 40, 694018
```

> ssh pi_user@pi_n

PC   SSH
    CLIENT

For bulk ope
- write list of pi's ip
- use xarg
- can be over WIFI, r
over internet

```
program.")
```

→Arduino

Example of sending data via USB

```
.Serial(USB_PORT, 9600, timeout=2)
```

While TRUE do:

Register licks & give water/food every n licks

```cpp
#include "Time.h"

void setup(){
  Serial.begin(9600);
  setTime(12,0,0,1,1,11);
}

void loop(){
  Serial.print("I'm sending a message at ");
  Serial.print(hour());
  printDigits(minute());
  printDigits(second());
  Serial.println();
  delay(5000);
}

void printDigits(int digits){
  Serial.print(":");
  if(digits < 10)
    Serial.print("0");
  Serial.print(digits);
}
```

```python
line = usb.readline()  # read input from Arduino
```

With this we can control arduino via USB w

```cpp
void loop() {
    // Read and execute commands from serial port
    if (Serial.available()) {  // check for incoming seria
        String command = Serial.readString();  // read comm
serial port
        if (command == "led_on") {  // turn on LED
            digitalWrite(LED_BUILTIN, HIGH);
        } else if (command == "led_off") {  // turn off LED
            digitalWrite(LED_BUILTIN, LOW);
        } else if (command == "read_a0") {  // read and se
value
            Serial.println(analogRead(A0));
        }
    }
}
```

Pi

SSH
SERVER

3)

umber_n 'python3 /home/user/scripts/.../usb.py
I

frame + time stamp
arduino sensor read + time stamp

erations

more difficult

WIth this we put a timestamp (in ms precision) to each output
read of sensors

```c
#define _POSIX_C_SOURCE 200809L

#include <inttypes.h>
#include <math.h>
#include <stdio.h>
#include <time.h>

void print_current_time_with_ms (void)
{
    long            ms; // Milliseconds
    time_t          s;  // Seconds
    struct timespec spec;

    clock_gettime(CLOCK_REALTIME, &spec);

    s  = spec.tv_sec;
    ms = round(spec.tv_nsec / 1.0e6); // Convert nanoseconds to milliseconds
    if (ms > 999) {
        s++;
        ms = 0;
    }

    printf("Current time: %"PRIdMAX".%03ld seconds since the Epoch\n",
            (intmax_t)s, ms);
}
```

ith the pi

al data
nand from

)

nd A0 analog