core1 1 1 1 1 1 1 1 1 1 1

$2^9\ 2^8\ 2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

512 256 128 64 32 16 8 4 2 1

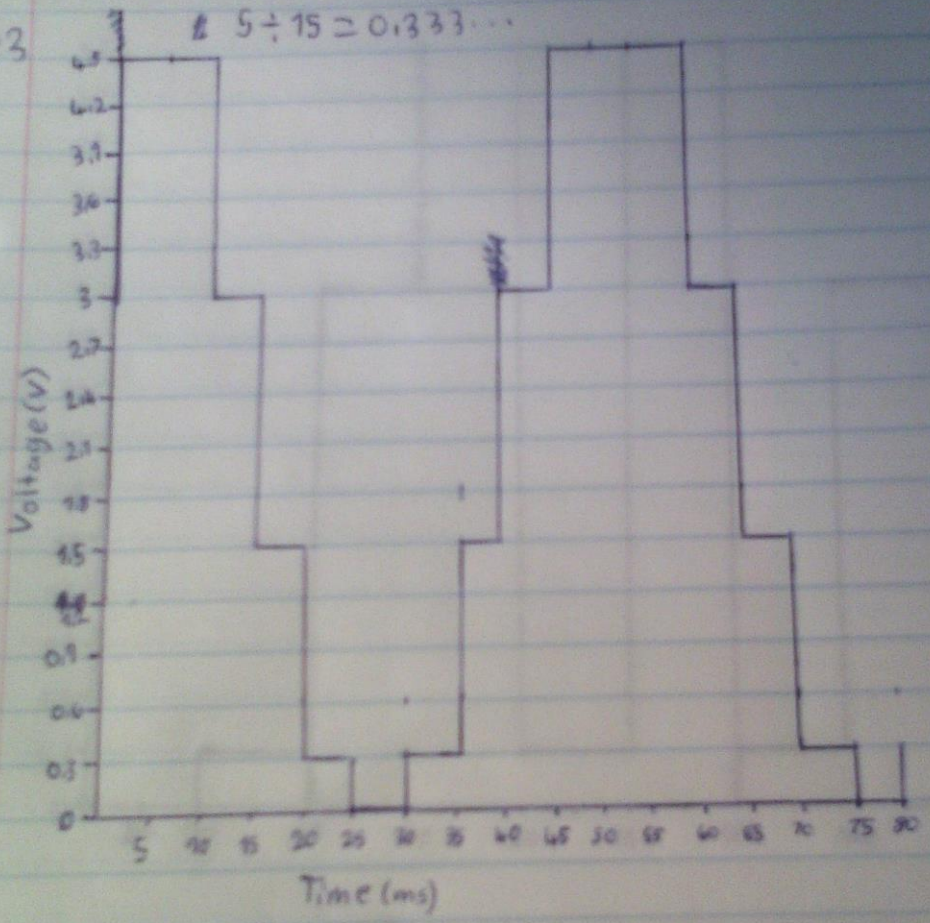
512+256+128+64+32+16+8+4+2+1 = 891 different values

core2

5 ÷ 891 = 0.0056 V = 6 milli volts

core3

5 ÷ 15 = 0.333...



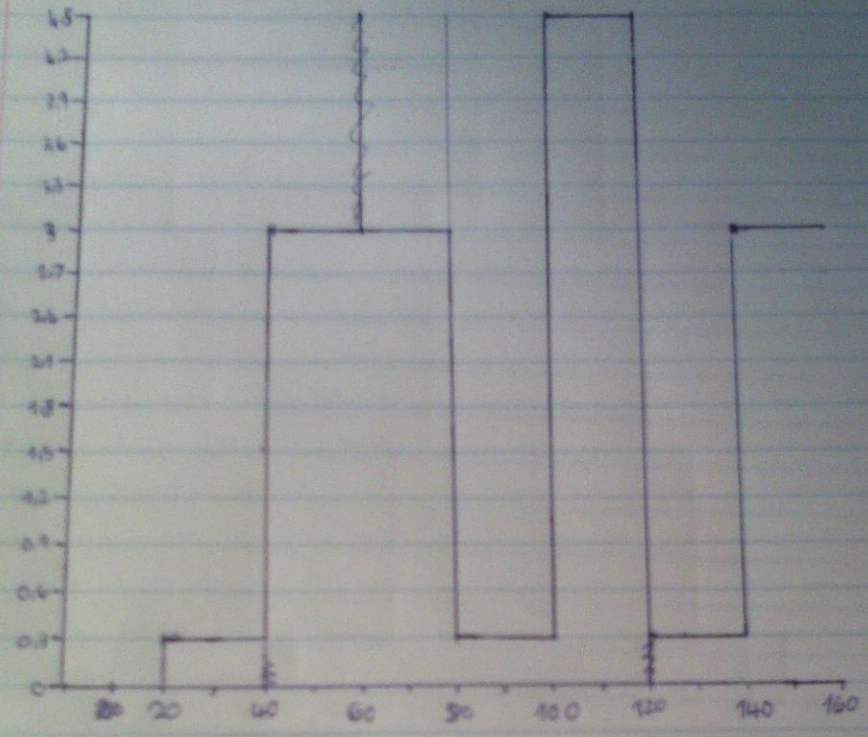
ore4

T = 50 ms = 0.05 s

1 x 0.05 = 20 waves per second

∴ f = 20 Hz

5



the sampling rate should be at least twice the frequency of the signal

Signal to noise ratio = $\frac{\text{min signal amplitude}}{\text{max noise amplitude}}$

$\frac{0.5}{0.75} = 0.667$

if the signal to noise ratio is >1 then the noise is almost non-existent relative to the signal but the closer the raito is to 0 the harder it is to distinguish the signal from the noise

```
#include <stdio.h>
#include <time.h>


extern "C" int InitHardware();
extern "C" int ReadAnalog(int ch_adc);
extern "C" int Sleep(int sec, int usec);

int main(){
    InitHardware();
    int adc_reading;
    adc_reading = ReadAnalog(0);
    printf("%d\n", adc_reading);
    Sleep(1,000000);
    adc_reading = ReadAnalog(2);
    printf("%d\n", adc_reading);
    Sleep(1,000000);
    adc_reading = ReadAnalog(4);
    printf("%d\n", adc_reading);
    Sleep(1,000000);

    return 0;
}
```

completion 2

```
recived
4
3
0
```

completion 3

core 9

```
#include <stdio.h>
#include <time.h>


extern "C" int InitHardware();
extern "C" int ReadAnalog(int ch_adc);
extern "C" int Sleep(int sec, int usec);

int main(){
    InitHardware();
    int adc_reading;
    int i = 0;
    while(i <= 5){
        adc_reading += ReadAnalog(0);
        printf("%d\n", adc_reading);
        i += 1;
        Sleep(0,500000);
    }
    if(i = 6){
    adc_reading /= 5;
    printf("%d\n", adc_reading);
    }
    return 0;
}
```

completion 4

```
#include <stdio.h>
#include <time.h>


extern "C" int InitHardware();
extern "C" int ReadAnalog(int ch_adc);
extern "C" int Sleep(int sec, int usec);

int main(){
    InitHardware();
    int adc_reading;
    int min = 0;
    int max = 10000000000000000;
    int reading = 0;
    int halfRange = 0;
    int i = 0;
    while(i < 5){
        reading = ReadAnalog(0);
        adc_reading += reading;
        if(reading > max){
            max = reading;
        }
        if(reading < min){
            min = reading;
        }
        printf("%d\n", "reading"+adc_reading);
```

```
        i += 1;
        Sleep(0,500000);
    }
    if(i = 6){
        adc_reading /= 5;
        printf("%d\n", "avrage reading"+adc_reading);
        halfRange = max - min;
        halfRange /= 2
        printf("%d\n", "half range"+adc_reading);
    }
    return 0;
}
```