

Programmable Drawing Robot

Michael Sekatchev

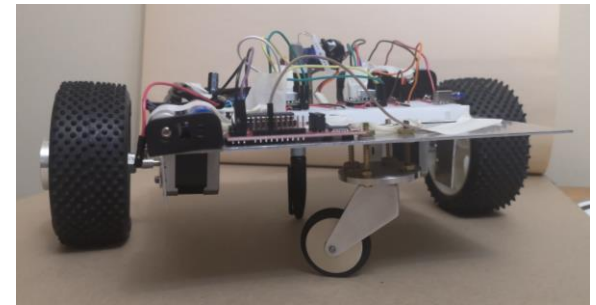
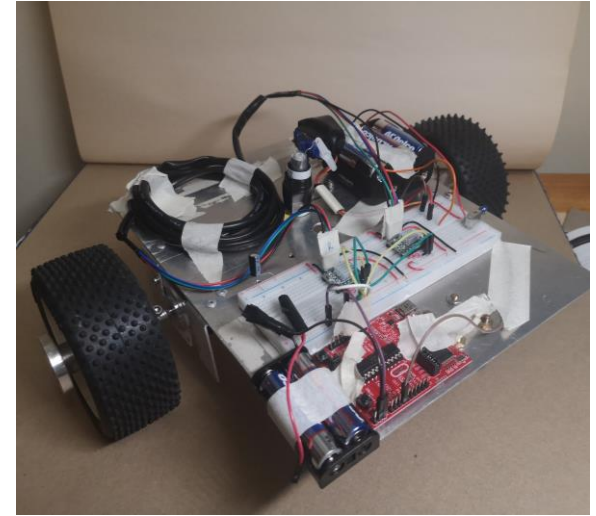
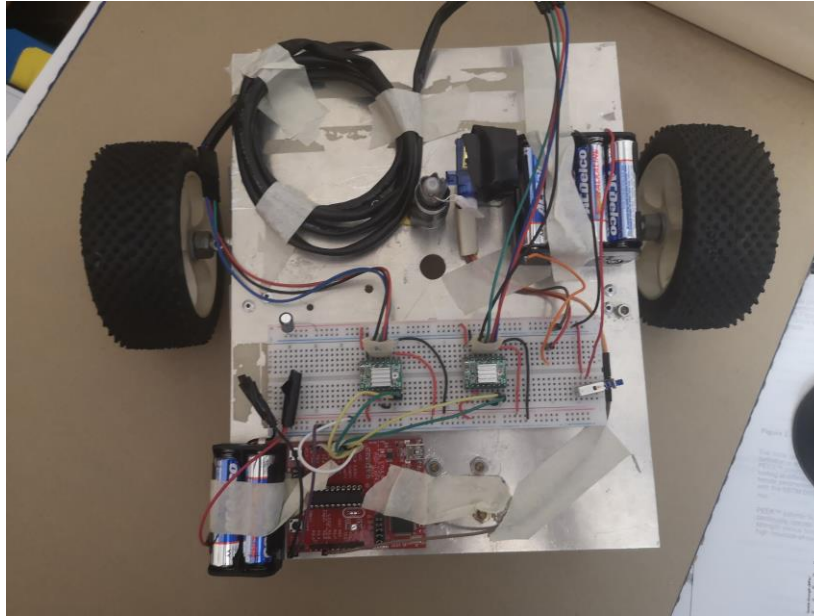
3rd year Honours Physics

PHYS 319 Final Project

04-04-2022

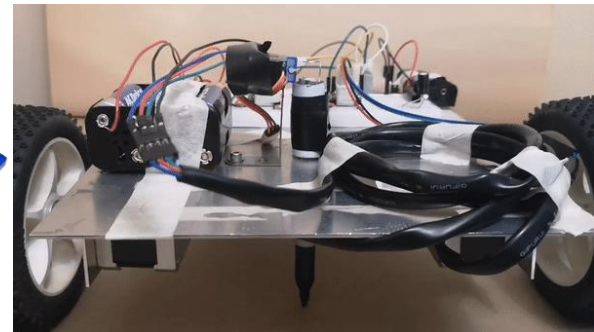
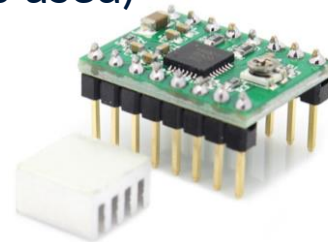
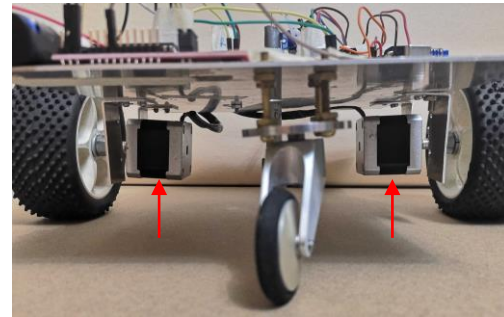
Project Introduction

- Robot that draws a pre-programmed pattern on a flat surface using a Sharpie
- Can draw separate lines by lifting the sharpie off the surface



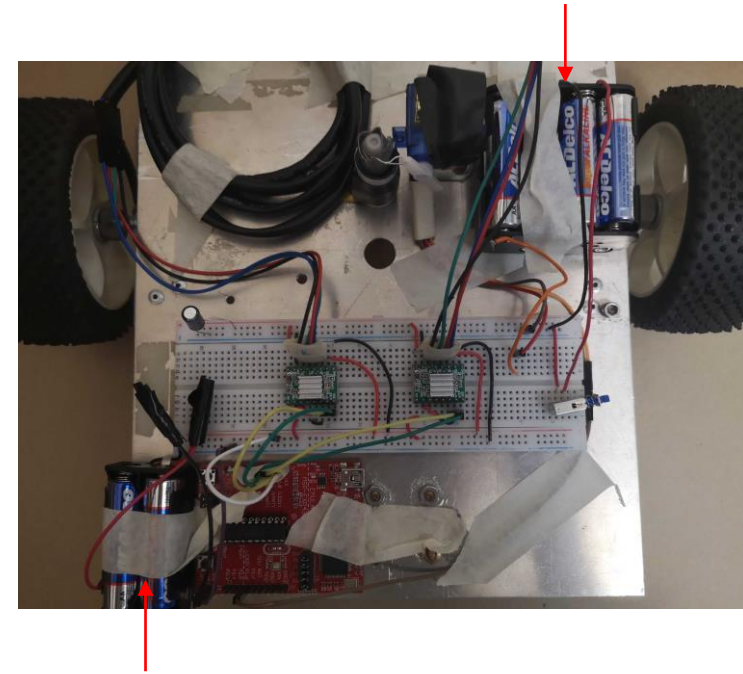
Hardware details

- 2 NEMA 17 **stepper motors** driving each wheel, and a dummy caster wheel
 - NEMA 17 steppers are powered with 8-35 V
 - Controlled by pulses sent via A4988 stepper motor drivers, powered by 3.3 V MSP430 logic (3 V is used)
 - 2 pins for control: DIR for direction, STEP for sending pulses
 - Microstep resolution controlled via 3 pins (set to 1/16th step for highest accuracy)
- A **servo motor** controls raising/lowering of the Sharpie on the drawing surface
 - Powered with 5 V (4.5 V is used)
 - Position angle set via period of PWM signal (3.3 V)

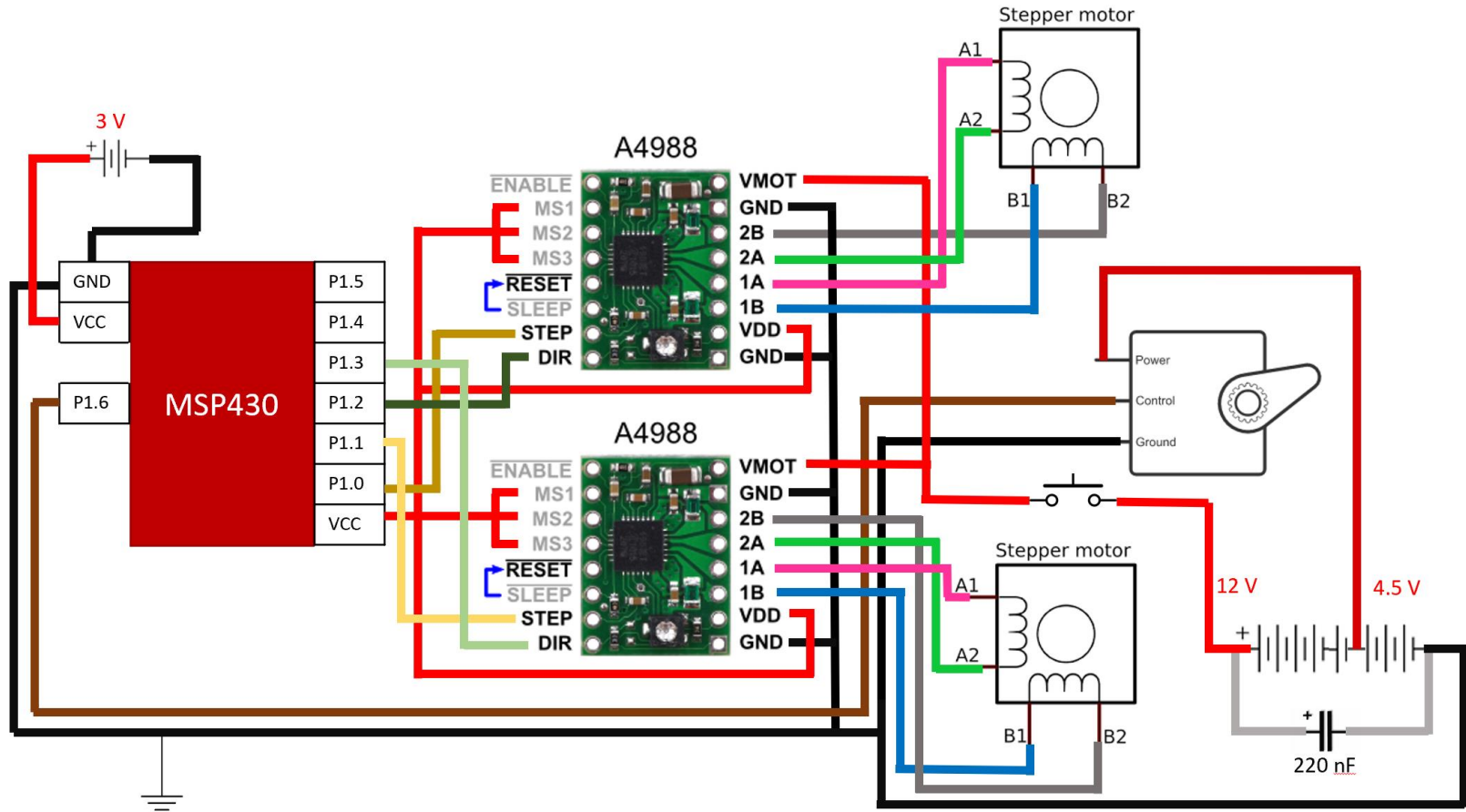


Circuitry details

- Uses 3 different power sources from 2 battery packs, all using AA batteries
 - **12 V** battery pack powers stepper motors through drivers
 - 220 nF capacitor used to stabilize voltage
 - Pushbutton switch used to turn steppers ON/OFF
 - **3 V** battery pack powers MSP430 logic, servo, and stepper driver control
 - **4.5 V** is split off 12 V battery pack and used to power servo
- Details in circuit diagram on next slide



Circuitry details



Software details

- Programmed in a systematic way via composition of 5 functions
- Code is indented to be legible to easily program to draw any desired doodle
- Main idea: **control direction and duration of the movement of each stepper motor by sending short pulses to the drivers**

d(delay_time)

Delay by
delay_time/10
milliseconds

**run(r_step, l_step,
r_dir, l_dir)**

Sends 1 pulse to the
specified steppers' STEP
and the specified
direction DIR using P10UT
and d()

**run_time(r_step, l_step,
r_dir, l_dir,
t)**

Calls run() t times, waiting
for 50 ms between calls

turn(dir, deg)

Calls run_time to turn
steppers in opposite
directions to turn left or
right by deg degrees*

go(dir, duration)

Calls run_time
duration times to turn
steppers in same
directions to move
forward or backward

*Robot had to be calibrated to match degrees to number of stepper motor microsteps

Software details – example

- Let's draw the letter M

```
CCR1 = DOWN;
```



Software details – example

- Let's draw the letter M

```
CCR1 = DOWN;
```

```
go(FWD, 1200);
```



Software details – example

- Let's draw the letter M

```
CCR1 = DOWN;
```

```
go(FWD,1200);
```

```
CCR1 = UP;
```

```
turn(LEFT,DEG15);
```



Software details – example

- Let's draw the letter M

```
CCR1 = DOWN;
```

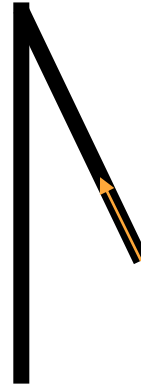
```
go(FWD,1200);
```

```
CCR1 = UP;
```

```
turn(LEFT,DEG15);
```

```
CCR1 = DOWN;
```

```
go(BAC,900);
```



Software details – example

- Let's draw the letter M

```
CCR1 = DOWN;
```

```
go(FWD,1200);
```

```
CCR1 = UP;
```

```
turn(LEFT,DEG15);
```

```
CCR1 = DOWN;
```

```
go(BAC,900);
```

```
turn(RIGHT,DEG30);
```



Software details – example

- Let's draw the letter M

```
CCR1 = DOWN;
```

```
go(FWD,1200);
```

```
CCR1 = UP;
```

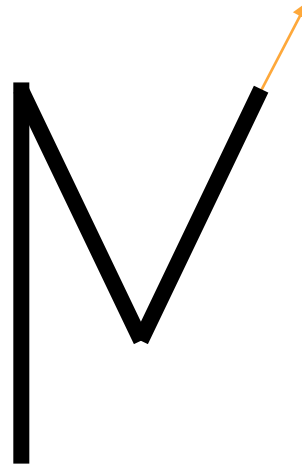
```
turn(LEFT,DEG15);
```

```
CCR1 = DOWN;
```

```
go(BAC,900);
```

```
turn(RIGHT,DEG30);
```

```
go(FWD,900);
```



Software details – example

- Let's draw the letter M

```
CCR1 = DOWN;
```

```
go(FWD,1200);
```

```
CCR1 = UP;
```

```
turn(LEFT,DEG15);
```

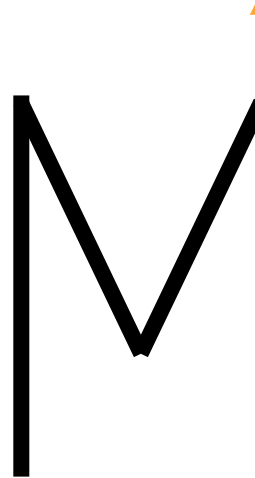
```
CCR1 = DOWN;
```

```
go(BAC,900);
```

```
turn(RIGHT,DEG30);
```

```
go(FWD,900);
```

```
turn(LEFT,DEG15);
```



Software details – example

- Let's draw the letter M

```
CCR1 = DOWN;
```

```
go(FWD,1200);
```

```
CCR1 = UP;
```

```
turn(LEFT,DEG15);
```

```
CCR1 = DOWN;
```

```
go(BAC,900);
```

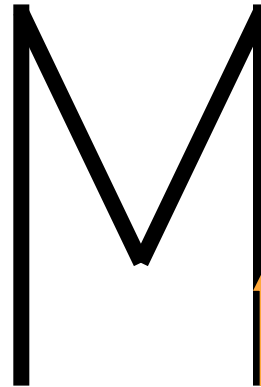
```
turn(RIGHT,DEG30);
```

```
go(FWD,900);
```

```
turn(LEFT,DEG15);
```

```
go(BAC,1200);
```

```
CCR1=UP;
```



Demo!

- I programmed the robot to write my (nick)name in Russian (note N≠И)

МИША

(links to videos in case of failure)

[video 1](#)

[video 2](#)

Appendix: Functions

```
void d(int delay_time){
    int i=0;
    for (i=0; i<delay_time; i++){
        _delay_cycles(100);
    }
}
```

```
void run(int r_step, int l_step, int r_dir, int l_dir){
    int on = 0b00000000;
    int off = 0b00000000;
    if(r_step){
        on+= 0b00000001;
    }
    if(l_step){
        on+= 0b00000010;
    }
    if(r_dir!=R_F){
        on+= 0b00000100;
        off+= 0b00000100;
    }
    if(l_dir==L_F){
        on+= 0b00001000;
        off+= 0b00001000;
    }
    d(10);
    P1OUT = on;
    d(10);
    P1OUT = off;
}
```

```
void run_time(int r_step, int l_step, int r_dir, int l_dir, int t){
    int count = 0;
    for(count = 0; count<t; count++){
        run(r_step,l_step,r_dir,l_dir);
        d(500);
    }
}
```

```
void go(int dir, int duration){
    if(dir == FWD){
        run_time(1,1,R_F,L_F,duration);
    }
    else{
        run_time(1,1,R_B,L_B,duration);
    }
}
```

```
void turn(int dir, int deg){
    if(dir == LEFT){
        run_time(1,1,R_F,L_B,deg);
    }
    else{
        run_time(1,1,R_B,L_F,deg);
    }
}
```