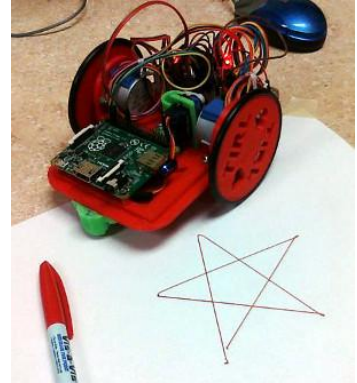


Open Source Turtle Robot

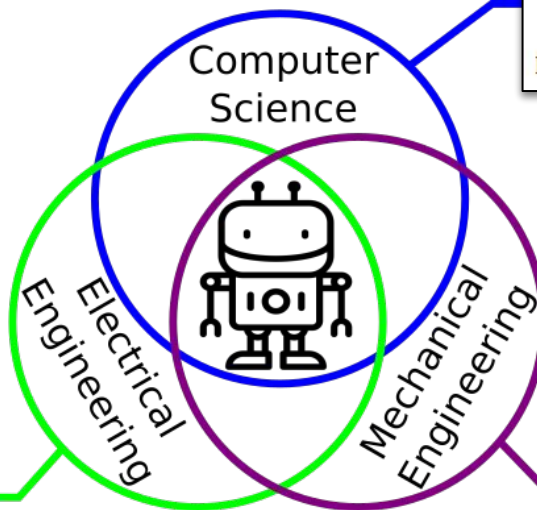
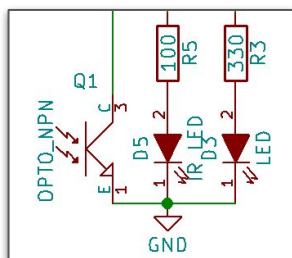
- **Robotics** is the exciting intersection of a number of engineering fields including mechanical engineering, electrical engineering, and computer science. This project was designed to make learning about these fields accessible and exciting.
- **Open Source** means every aspect of its design from its 3D parts, electronics, and software are available for study and modification, making it easy to build, modify, and improve.
- **Turtle robots** are controlled by simple instructions like **forward**, **backward**, **left**, and **right**, and their visual tracks are instructive as well as entertaining. They also demonstrate how systems with simple rules can have complex behaviors, something we see in nature all the time.



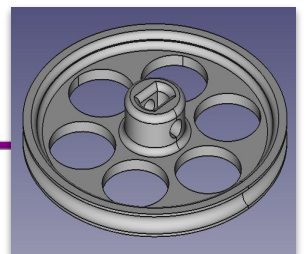
WHAT IS A ROBOT? ROBOTS ARE MACHINES THAT ARE PROGRAMMED TO REACT TO AND MANIPULATE THEIR ENVIRONMENTS.

WHICH OF THE FOLLOWING WOULD YOU CONSIDER TO BE A ROBOT?

- ☐ MARS ROVER
- ☐ 3D PRINTER
- ☐ AUTONOMOUS CAR
- ☐ ARTIFICIAL ARM



```
from turtle import *  
  
pendown()  
for x in range(5):  
    forward(100)  
    right(144)  
penup()
```



Going Further:

- Build a Turtle Robot of your own from Instructables: <http://bit.ly/OSTurtle>
- Play with Turtle graphics online at :
 - <https://blockly-games.appspot.com/> (block programming)
 - <http://bit.ly/ttturtle> (JavaScript)
 - <https://groklearning.com/hoc/activity/snowflake/> (Python)
- Do an “Hour of Code” activity at <https://hourofcode.com/us/learn>
- Check for Maker Spaces or programs in your community or at the library.
- ChickTech Workshops: <https://chicktech.org/>



Turtle Robot Datasheet

#TurtleRobot @ChickTechOrg @TheMakersBox

Robot Name: _____

Builder's Name: _____

Adafruit ItsyBitsy:

- Cortex M4 @ 120 MHz
- 512KB Flash/ 192KB RAM
- Runs Python!

Motion: 2 x 28byj 5 volt, Uni-polar Stepper Motors with 1:64 Gearing

Pen Control: 9g Servo

Output: LEDs

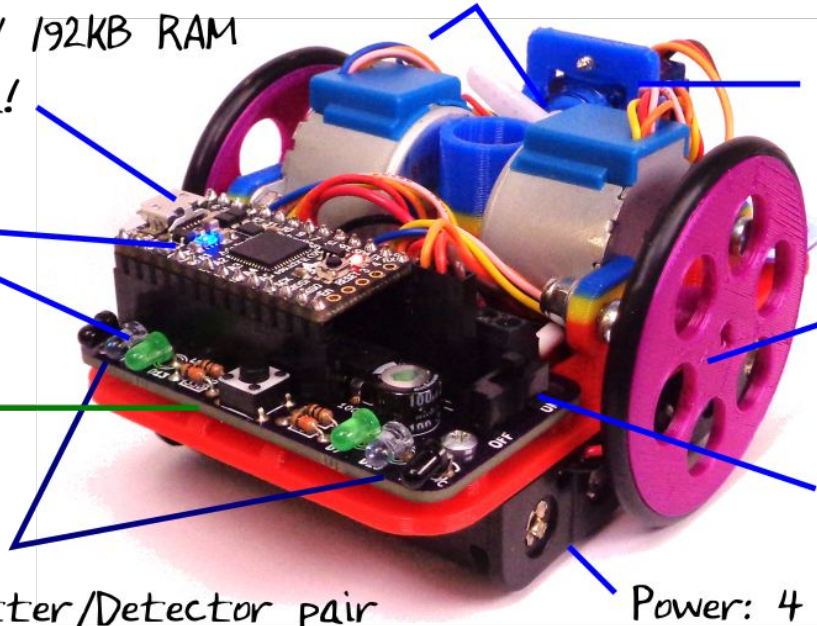
3D Printed Chassis and wheels

Input: Push button

Power Switch

Sensing: 2x IR Emitter/Detector pair

Power: 4 x AA (6 Volts)



Turtle Commands:

```
left(degrees)
right(degrees)
forward(distance) mm
backwards(distance) mm
penup()
pendown()
goto(x, y)
circle(radius, sides)
RGBled((R,G,B))
isButtonPushed()
```

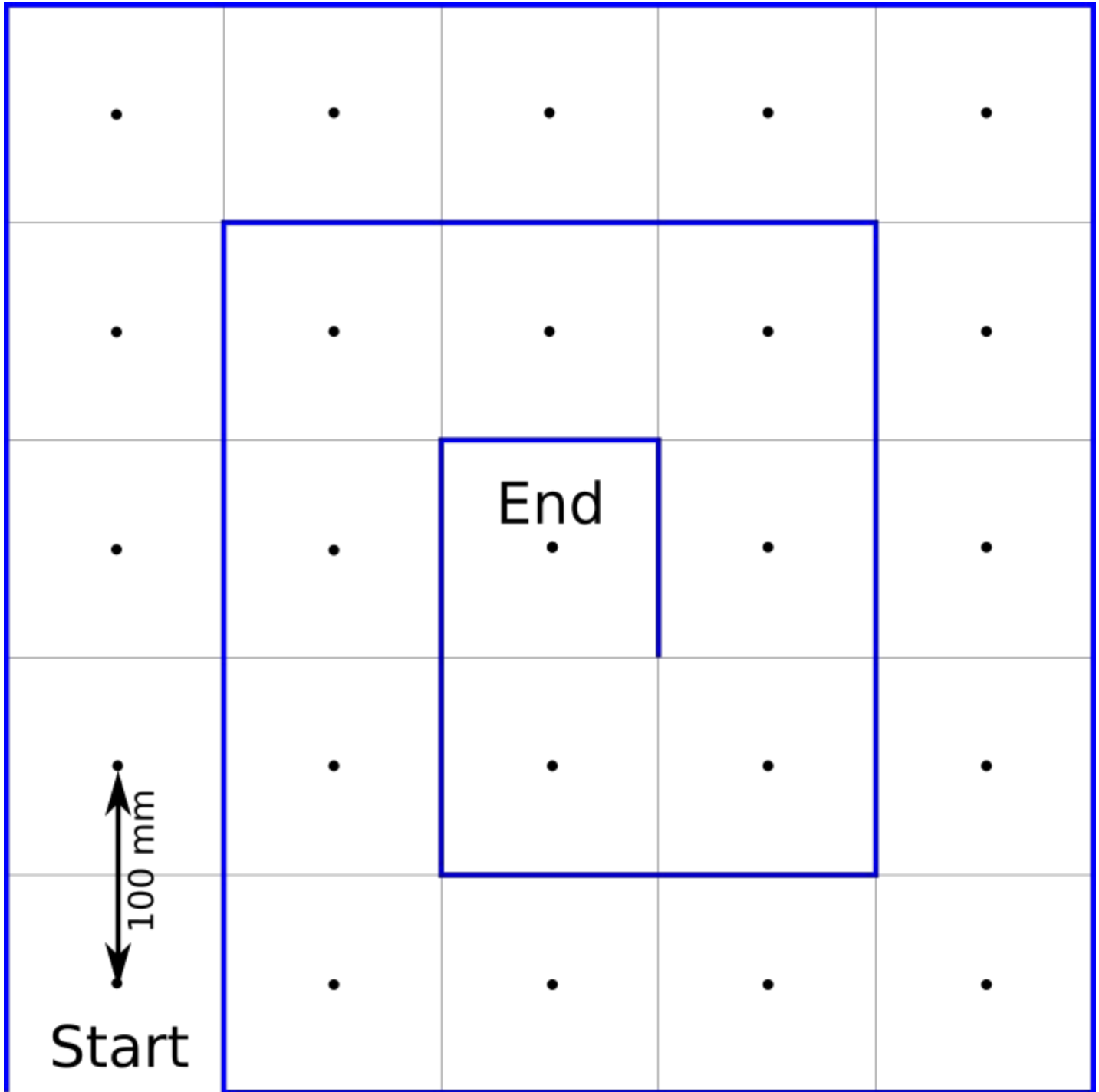


Calibration Parameters:

wheel_dia (mm): _____
wheel_base (mm): _____
PEN_UP (angle): _____
PEN_DOWN (angle): _____



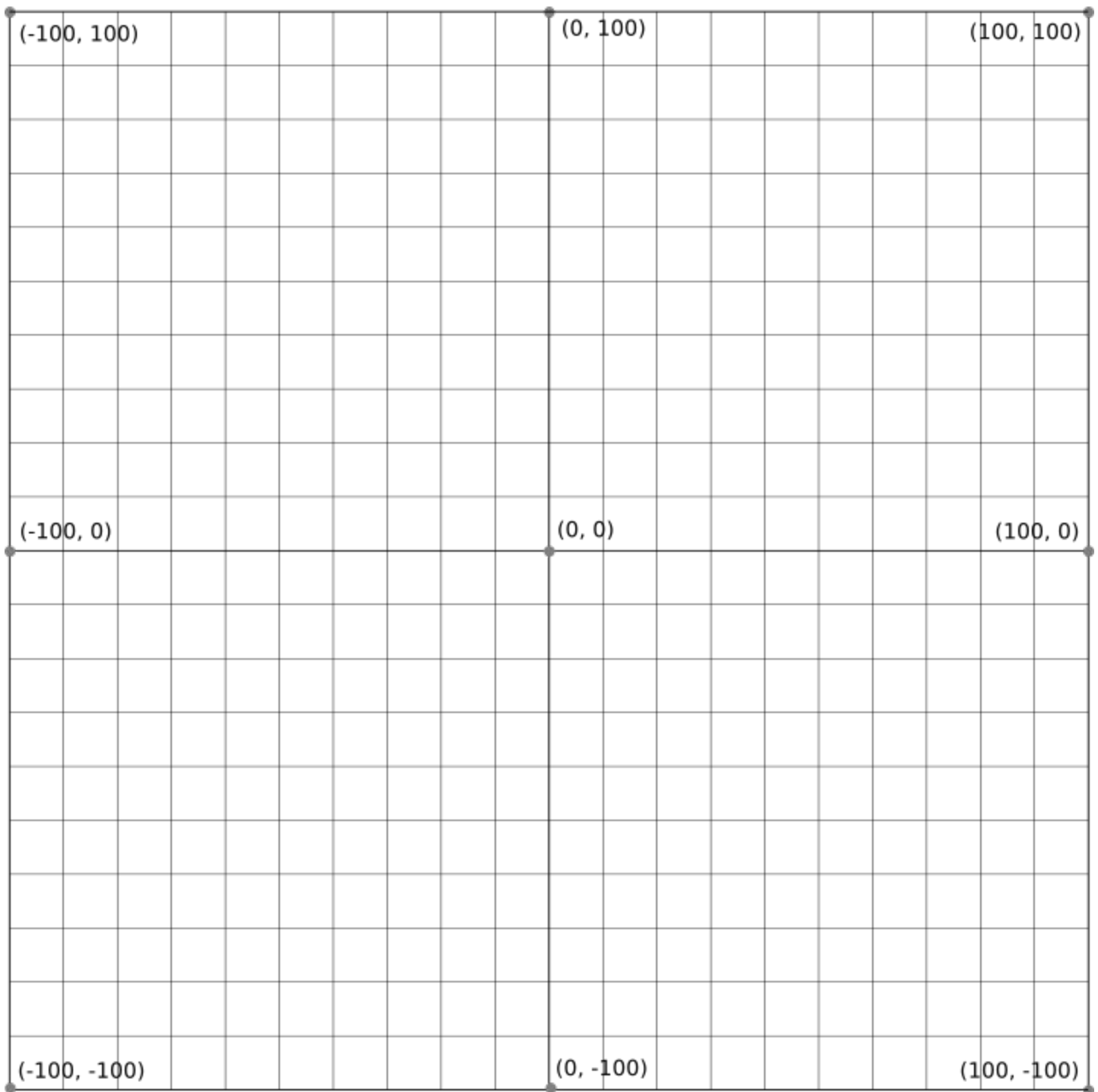
Full details at <http://bit.ly/OSTurtle>



Complete the sequence to get your robot from Start to End. Bonus for using loops.

```
forward(400)  
right(90), ...
```

Turtle goto(x, y) Command



The **turtle goto()** command will move the turtle to the x, y coordinate. If pen is down, it will draw a line. If you put the points in a list, you can iterate through them. What shape does the following make? Can you make your own shape above?

```
points = [(0, 0), (-8, -8), (-4, -24), (-16, -36), (-28, -28), (-36, -32), (-24, -44),  
          (-28, -60), (-20, -76), (-32, -88), (-24, -96), (-16, -84), (0, -92),  
          (16, -84), (24, -96), (32, -88), (20, -76), (28, -60), (24, -44), (36, -32),  
          (28, -28), (16, -36), (4, -24), (8, -8), (0, 0)]
```

```
for point in points:  
    goto(point)
```


SAFETY PLAN

Hot Glue:

- Is **hot**. It burns. Remove it from skin immediately and run water over it. Contact an instructor.



Soldering & Wire cutting:

- **Always wear safety glasses when soldering or cutting wires.** Eyes don't grow back.
- Keep long hair in ponytail holder. Burning hair stinks!
- Run water over burns and contact instructor.
- Wash hands after soldering.

Cuts:

- Contact instructor.
- Wash, neosporin, band aid.

Ken's 3Ps for Problem Solving:

- **Be Patient**, ask a friend for help first.
- **Be Persistent**, it can take many tries.
- **Be Positive**, you can figure it out!

Challenge Activities:

- From the snowflake tutorial, create your own custom snowflake and put the code in *my_snowflake.py* template
- In snowflake examples, use the function *random.randint(low, high)* to generate random number of arms, angles, and/or lengths.
- In *turtle_obstacles.py*, the turtle only stops when sensing a wall. Write code to have turtle navigate away from obstacles it senses.
- Try the different songs in *rttt_example.py* to find your favorite.
- Open *music_example.py* to see how tones are played. Can you adapt your favorite tune in code?
- What does *turtle_goto_example.py* draw? Use the handout to create your own object.
- There is a *isButtonPushed()* function that returns **True** if pushed, **False** if not. What could you use that for? Hint: look in *turtle_eye_check.py*!
- Bonus level: Since turtle can see and emit IR, with a friend could you create a predator / prey pair, with one seeking IR, and the other fleeing IR?