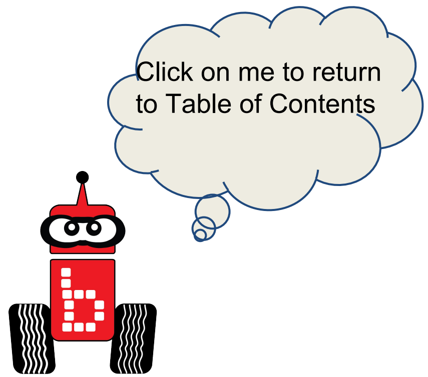
Motor Position Counters

- **Key Concepts**
  - **Understand what motor position counters are and how to use them.**
- **Pacing**
  - **Over several class periods.**

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Table Of Contents



KISS INSTITUTE FOR PRACTICAL ROBOTICS

Standards

Goal:

- Students will familiarize themselves with the functions `msleep()` and `motor()`
- Students will understand how to move their robots in the following manner: forwards, backwards, straight,, circles, right and left turns

Standards:

Common Core State Standards Math Practices

CCSSMP1: Make sense of problems and persevere in solving them

CCSSMP2: Reason abstractly and quantitatively

CCSSMP4: Model with mathematics

CCSSMP6: Attend to precision

CCSSMP8: Look for and express regularity in repeated reasoning

Next Generation Science and Engineering Practice

1: Asking questions and defining problems

2: Developing and using models

3: Planning and carrying out investigations

4: Analyzing and interpreting data

5: Using mathematics and computational thinking

6: Constructing explanations and designing solution

7: Engaging in argument from evidence obtaining, evaluating, and communicating information

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Standards Continued

Standards Continued:

2016 ISTE Standards

Empowered Learner

1c: Students use technology to seek feedback that informs and improves their practice and to demonstrate their learning in a variety of ways.

1d: Students understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

Knowledge Constructor

3d: Students build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions.

Innovative Designer

4a: Students know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems.

4b: Students select and use digital tools to plan and manage a design process that considers design constraints and calculated risks.

4c: Students develop, test and refine prototypes as part of a cyclical design process.

4d: Students exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.

Computational Thinker

5a: Students formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.

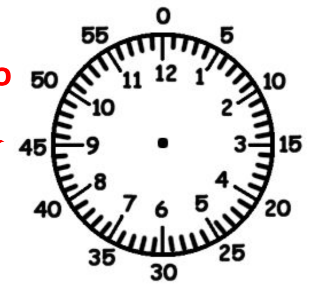
KISS INSTITUTE FOR PRACTICAL ROBOTICS

Understanding Motor Position Counter
Activity 1

Read and discuss this slide.

Each motor used by your robot has a built-in motor position counter, which you can use to calculate the distance traveled by the robot!

Similar to how a clock is divided into 60-second intervals (ticks).



- Motor position is measured in "**ticks**".
- Your motors have ***approximately*** **1400 ticks per *revolution*** (This is not related to the size of your wheel).
- We need a new function for ticks.

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Moving at Velocity

Move at Velocity (mav)

We are going to now use a function that is much more accurate.

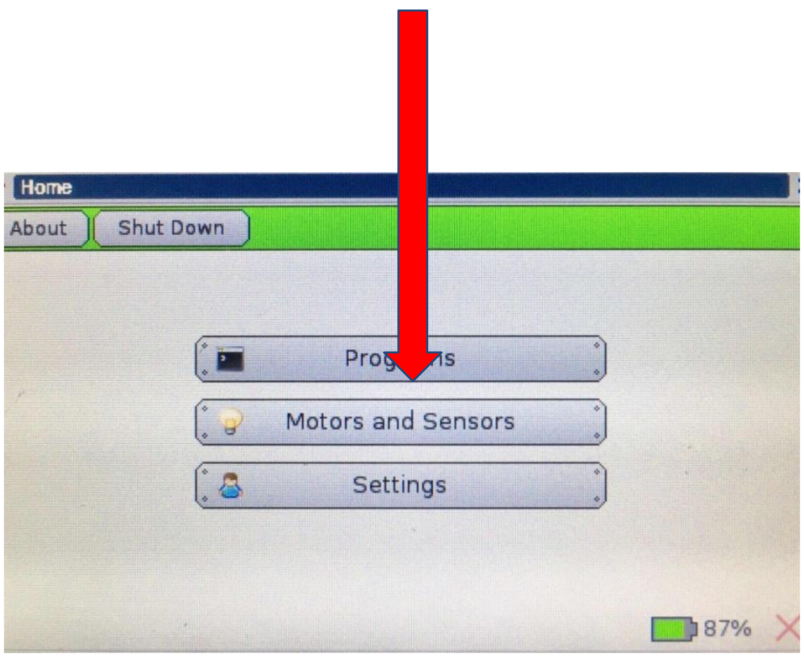
mav (port #, ticks per second);

mav (0, 1000);

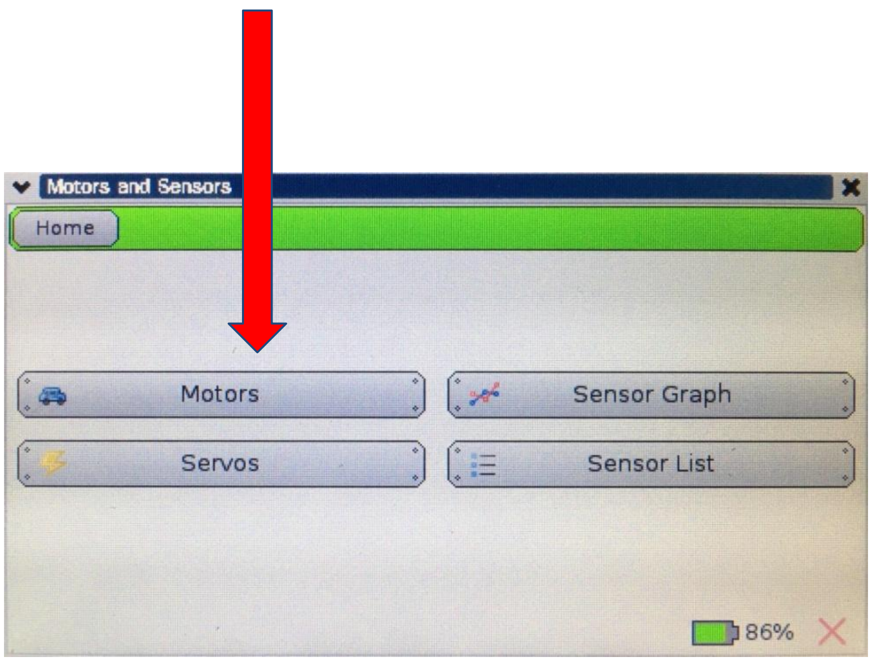
- ticks per second values are -1500 to 1500
- maximum and minimum do the same thing, count ticks per second.

Using the Robot Motor Screen to See Motor Positions Activity 2

1. Select Motors and Sensors



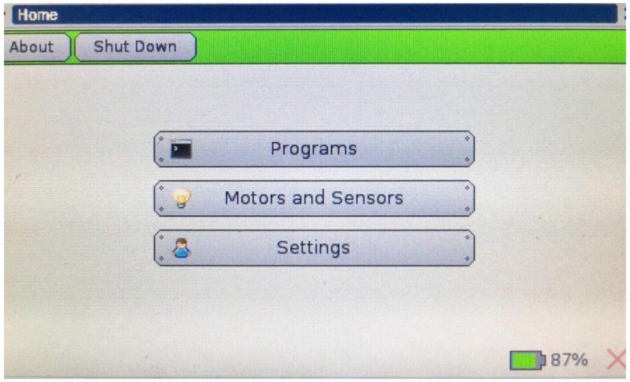
2. Select Motors



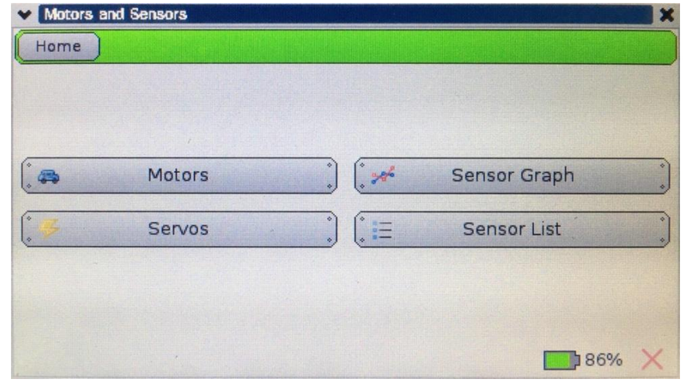
KISS INSTITUTE FOR PRACTICAL ROBOTICS

Using the Robot Motor Screen to See Motor Positions

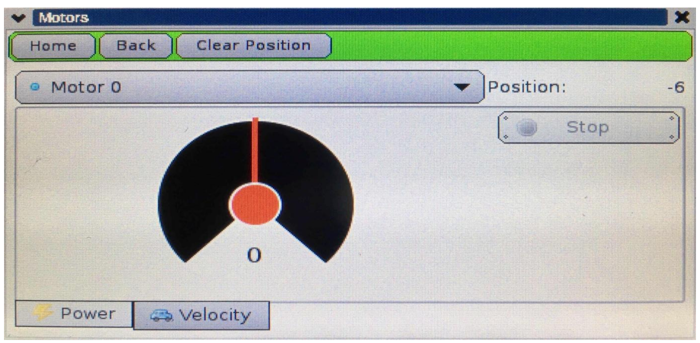
1. Select Motors and Sensors



2. Select Motors



3. Select Motor Port (allows you to select the motor of your choice)



4. Make sure Velocity is selected

5. Clears position of motor

Motor Position is in "Ticks"

6. Clear the motor position and then using your hand to rotate the robot's wheel, plugged into port 0, and watch the position counter. What happens if you turn the wheel in the opposite direction?

Place your robot on a surface and roll it forward to measure the # ticks from a starting position to another location or object.

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Functions:

mav (port #, ticks per second); **//move at velocity**

```
mav (0,1000);
```

- ticks per second values are -1500 to 1500
- maximum and minimum do the same thing, count ticks per second.

```
get_motor_position_counter(3);
```

- Tells the robot the number of ticks the motor on port #3 has rotated.

```
clear_motor_position_counter(3);
```

- Resets the robot's tick counter to 0 for the motor on port #3.

Acronyms:

cmpc= clear motor position counter

gmpc= get motor position counter

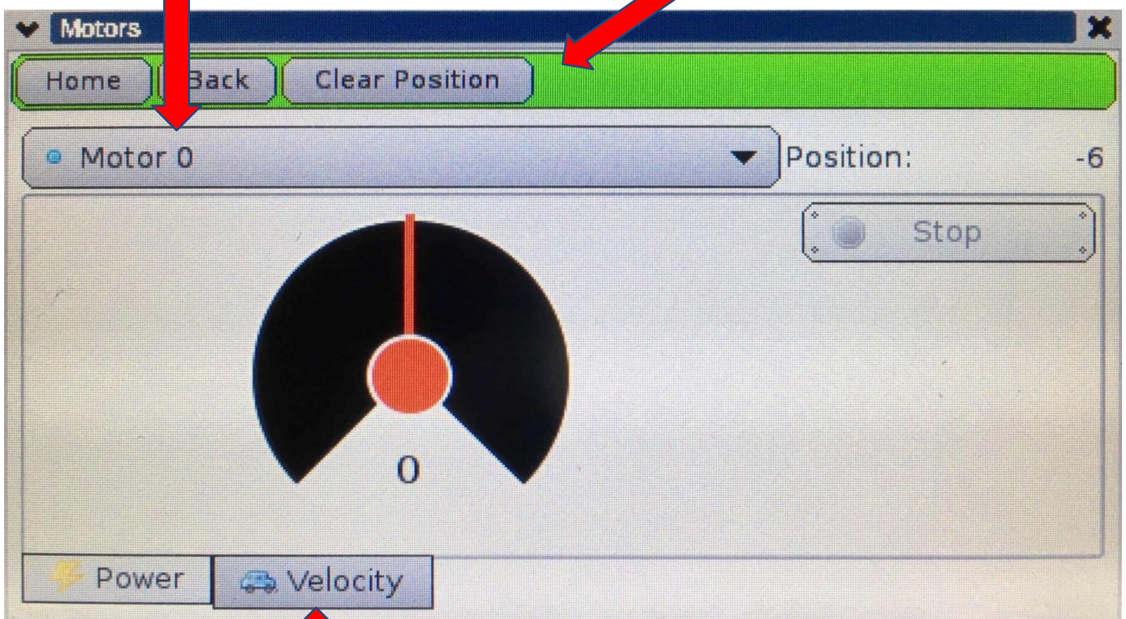
KISS INSTITUTE FOR PRACTICAL ROBOTICS

Using the Robot Motor Screen to See Motor Positions Activity 2

3. Select Motor Port (allows you to select the motor of your choice)

5. Clears position of motor

Motor Position in "Ticks"



4. Make sure Velocity is selected

6. Clear the motor position and then using your hand to rotate the robot's wheel, plugged into port 0, and watch the position counter. What happens if you turn the wheel in the opposite direction

Place your robot on a surface and roll it forward to measure the # ticks from a starting position to another location or object

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Motor Position Counter
Activity 3

New functions:

Read and discuss new functions you use:

Motor Port #
(#0 – 3)

```
get_motor_position_counter(3);
```

- Tells the robot the number of ticks the motor on port #3 has rotated.

Motor Port #
(#0 – 3)

```
clear_motor_position_counter(3);
```

- Resets the robot’s tick counter to 0 for the motor on port #3.

Proceed to the next slide…...

Using Acronyms your Code

1. Turn to a partner and discuss your favorite acronyms or abbreviations!
2. Share out.
3. Would you use an acronym without knowing it's meaning or understanding?
4. Acronyms:

cmpc= clear motor position counter

gmpc= get motor position counter

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Using the Robot Motor Screen to See Motor Positions
Activity 2

1. Write a program that will move your robot to 2200 ticks.

Hints:

- Remember to tell both motors
- Clear your counter

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Motor Position Counter
Activity 3

1. Use code planning paper to plan the steps in your code to run one motor on your robot approximately one revolution.

Hint: You will be using a loop in your code

2. Write the code and run the program

Psuedocode sample on the next slide

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Motor Position Counter
Activity 3 - Psuedocode Sample

1. Clear the counter
2. Continuously get the motor counter for one motor
3. Power that motor
4. Stop

Sample code solution on next slide…...

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Using Motor Position Counter Functions
Activity 3 - Solution

```
#include <kipr/botball.h>

int main()
{
  clear_motor_position_counter(3);

  while (get_motor_position_counter(3) < 1400)
  {
    motor(3, 50);
  }

  ao();

  return 0;
}
```

How many *revolutions* will the motor rotate?

Intro to Motor Position Counters
Activity 4

Goal: Write a program that drives your robot forward for 4 *motor revolutions*, and then stops.

- Remember: there are 1400 motor ticks per revolution.

1. Use code planning paper to plan the steps in your code to move your robot forward approximately 4 revolution.
2. Write the code and run the program.

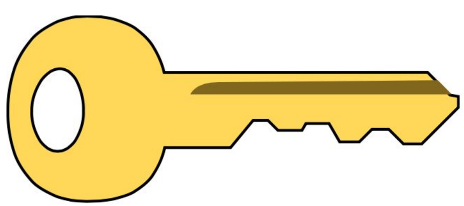
Hint: Your code will need a loop

1400 X 4 =?

Intro to Motor Position Counters
Activity 4 Continued

Pseudocode/Steps

1. Clear motor position
2. Continuously get the motor counter for two motors.
3. Move forward
4. Stop



Sample Code Solution

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Intro to Motor Position Counters
Activity 4 -

Have a class discussion for reason that you may want to use a counter?

<u>Some Possible Reasons</u>

- How could you calculate an exact distance in centimeters to travel? (**Hint:** Use **wheel circumference divided by 1400** to calculate distance!)

- How could you modify your program to drive a specific distance? (**Hint:** Consider writing a function with an argument for the distance.)

- How could you modify your program to accurately turn left or right?
- How far did the robot travel? Was it always the same?

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Ticks Per Revolutions - Activity 5

Math Extension

There are *approximately* 1400 ticks per revolution.

However, this number might be different for your robot.

Goal: Determine the *exact* number of ticks per wheel revolution for *your* robot.

Materials: Robot, Mat A, Dry Erase Marker, Meter Stick Notebook, Pencil

Prerequisite: Activity 4

Procedure:

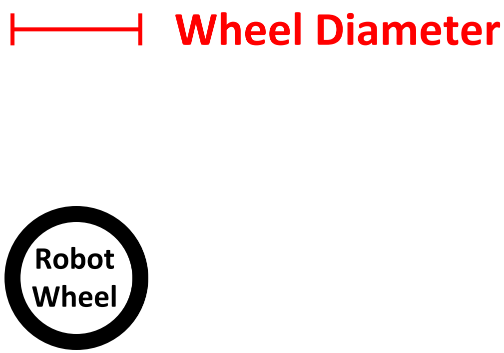
1. Place your robot behind the starting line on Mat A.
2. Use a dry erase marker to draw a small line on the mat where the center of the robot wheel touches the mat. Label this as “Start”. (Add figure)
3. Run your program from Activity 4.
   – By default, your program from Activity 4 should drive the robot forward 5600 ticks; if you changed this number in your code, that’s okay—we will write it down later.
4. When the robot stops, use a draw erase marker to draw a small line on the mat where the center of the robot wheel touches the mat. Label this as “End”.

Proceed to next slide...

Ticks Per Revolution:
Activity 5 Continued

Determine the following values, and write them down in your notebook.

1. **Wheel Diameter:** Measure the diameter (in centimeters) of your robot’s wheel, and write down the number.



Proceed to next slide...

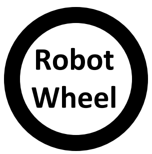
Ticks Per Revolution: Activity 5 Continued

Determine the following values, and write them down in your notebook.

2. **Wheel Radius:** Using the diameter, calculate the radius of your robot's wheel, and write down the number.

Wheel Diameter

Wheel Radius = Wheel Diameter / 2



Proceed to next slide...

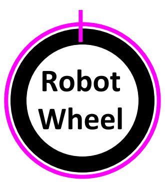
Ticks Per Revolution:
Activity 5 Continued

Determine the following values, and write them down in your notebook.

3. **Wheel Circumference:** Using the radius, calculate the circumference of your robot's wheel, and write down the number.

Wheel Diameter

Wheel Radius = Wheel Diameter / 2



Wheel Circumference = 2 × π × Wheel Radius

Proceed to next slide...

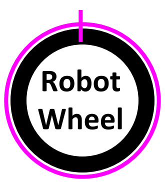
Ticks Per Revolution:
Activity 5 Continued

Determine the following values, and write them down in your notebook.

4. **Distance Traveled:** Measure the distance (in centimeters) between the “Start” and “End” points, and write down the number.

Wheel Diameter

Wheel Radius = Wheel Diameter / 2



Wheel Circumference = 2 × π × Wheel Radius

Distance Traveled

Proceed to next slide...

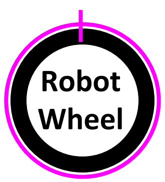
Ticks Per Revolution:
Activity 5 Continued

Determine the following values, and write them down in your notebook.

5. **# Revolutions:** Divide the distance traveled by the circumference of the wheel to determine the number of revolutions of the motor, and write down the number.

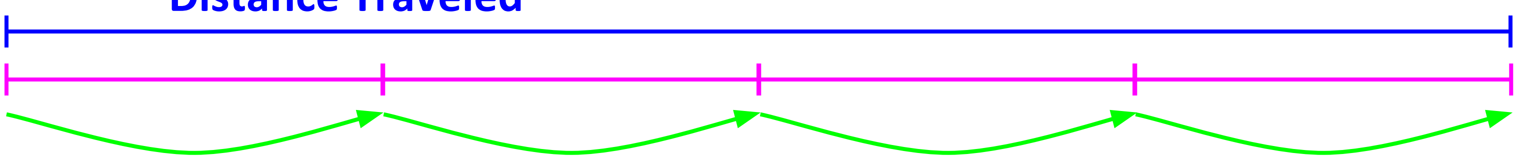
Wheel Diameter

Wheel Radius = Wheel Diameter / 2



Wheel Circumference = 2 × π × Wheel Radius

Distance Traveled



 # Revolutions = Distance Traveled / Wheel Circumference

Proceed to next slide...

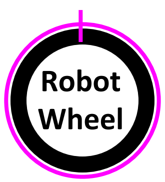
Ticks Per Revolution:
Activity 5 Continued

Determine the following values, and write them down in your notebook.

6. **# Ticks:** The number of ticks traveled when you ran your program from Activity 4.
   – By default, your program from Activity 4 should drive the robot 5600 ticks.

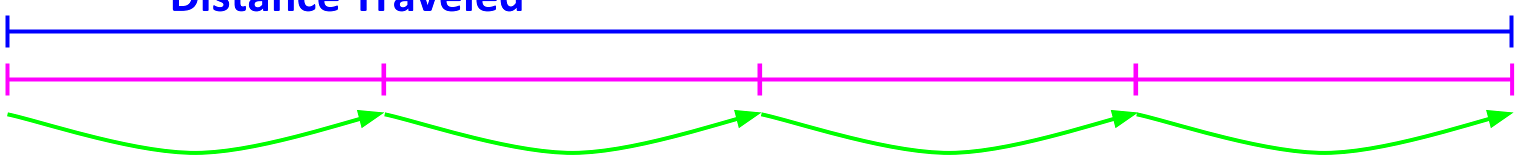
Wheel Diameter

Wheel Radius = **Wheel Diameter** / 2



Wheel Circumference = 2 × π × **Wheel Radius**

Distance Traveled



 # Revolutions = **Distance Traveled** / **Wheel Circumference**

```
while (get_motor_position_counter(0) < 5600)
```

 # Ticks From Activity 4

Proceed to next slide...

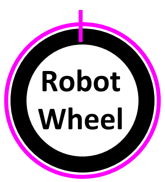
Ticks Per Revolution: Activity 5 Continued

Determine the following values, and write them down in your notebook.

7. **Ticks Per Revolution:** Divide the number of ticks traveled by the number of revolutions to determine the number of ticks per revolution.

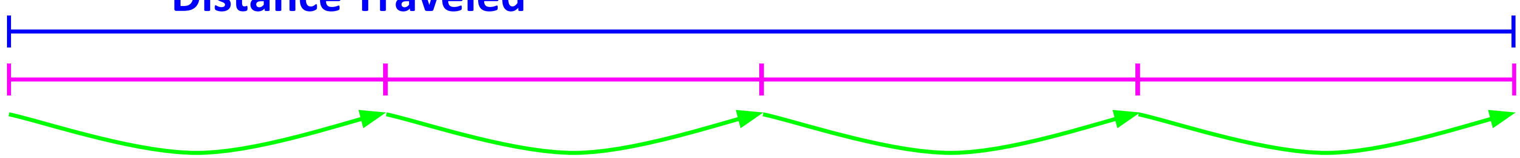
Wheel Diameter

Wheel Radius = Wheel Diameter / 2



Wheel Circumference = 2 × π × Wheel Radius

Distance Traveled



 # Revolutions = Distance Traveled / Wheel Circumference

```
while (get_motor_position_counter(0) < 5600)
```

 # Ticks From Activity 4

Ticks Per Revolution = # Ticks / # Revolutions

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Math Extension

Driving with Precision
Activity 6

Goal: Drive your robot forward *exactly* 30 centimeters.

Materials: Robot, Mat A, Dry Erase Marker, Notebook, Pencil, meter stick

Pre-requisite: Activity 4 and Activity 5

Preparation: Determine the following values, and write them down in your notebook (don't forget to include your units!).

1. **Goal Distance = 30 centimeters**
2. **# Revolutions = Goal Distance / Wheel Circumference**
3. **# Ticks = # Revolutions × Ticks Per Revolution**

From Activity 5

4. Modify the code from **Activity 4** by replacing the number of ticks (by default, 5600) with *your* **# Ticks** you just calculated.

Proceed to next slide...

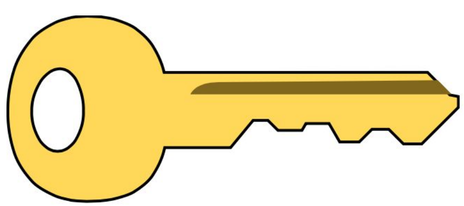
KISS INSTITUTE FOR PRACTICAL ROBOTICS

Driving with Precision
Activity 6

4. Modify the code from **Activity 4** by replacing the number of ticks (by default, 4000) with *your* **# Ticks** you just calculated.

```
while (get_motor_position_counter(0) < 5600) // is counter < # Ticks?
```

Replace this number with *your* **# Ticks**.



Sample Code Solution

Driving with Precision
Activity 6

Checking the Driving Precision

1. Place your robot behind the starting line on Mat A.

2. Use a dry erase marker to draw a small line on the mat where the center of the robot wheel touches the mat. Label this as “Start”.
   – **Note:** If you did this in Activity 5, you may use the same line.

3. Measure from the “Start” line you just drew to a point 30 centimeters forward. Draw a small line on the mat at this distance. Label this as “Goal”.

4. Run your modified program for this activity.

5. When the robot stops, use a draw erase marker to draw a small line on the mat where the center of the robot wheel touches the mat. Label this as “End”.

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Driving with Precision
Activity 6 - Reflections

- How close was the “End” line to the “Goal” line? Were they almost the same line?

- If they were *not* the same line, measure the distance between the “End” line and the “Goal” line, and write it down in your notebook.
  - Revisit **Activity 2** to improve your estimate of **Ticks Per Revolution**.

- How could you use motor position counters to move your robot around the mat?

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Assessments and Rubrics

Suggestions: *Understanding* rubric and or *Group Collaboration*