Activity M81

Task

Teacher will announce a set of numbers(within 0-1000) that the students will need to use to solve one and two step addition and subtraction problems via paper and pencil calculations. As soon as students have determined the result of the calculation, they will type it into the argument box on the mav () function of their stock program, compile the program, and run it. Process will be completed as many times as the teacher desires to reach fluency.

Presquite in Curriculum

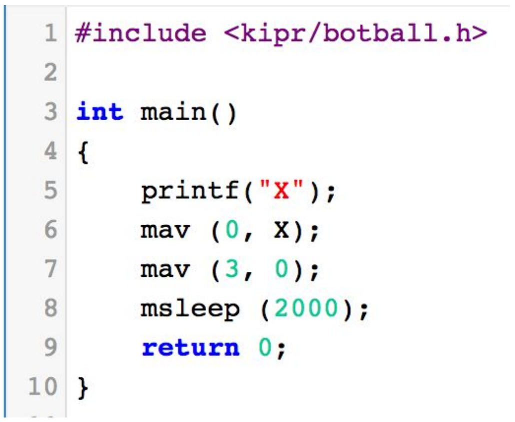
Completed Module 9- Moving Your Robot (understand using the mav () function)

Materials

- Assembled Robot
- JBC Mat A
- Paper and pencil

Planning

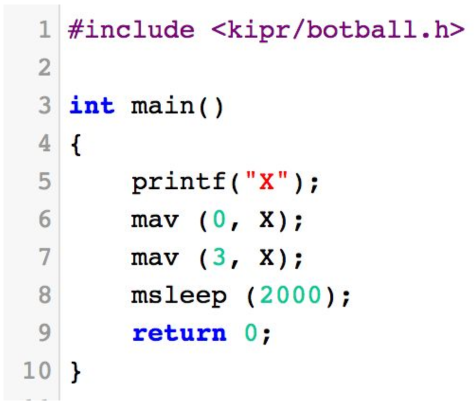
1. Each robot will be placed in the starting block of JBC mat A in an orientation so that forward straight motion would take the robot toward the other end of the mat.
2. Each student should start with the following stock program and fill in the "X" with their answer to the addition and/or subtraction problem that the teacher gives them. Additionally, students should put their answer to the problem in the printf function so that the answer can be verified on the robot's screen.



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

int main()
{
    printf("X");
    mav (0, X);
    mav (3, 0);
    msleep (2000);
    return 0;
}
```

This code will have the robot turn at the calculated speed



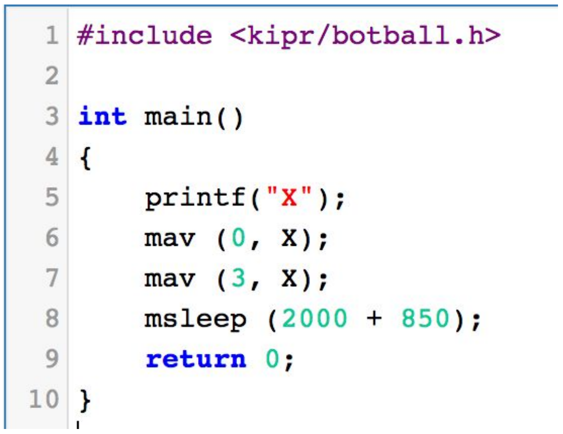
```
#include <kipr/botball.h>

int main()
{
    printf("X");
    mav (0, X);
    mav (3, X);
    msleep (2000);
    return 0;
}
```

This code will have the robot drive straight at the calculated speed

3. Instead of a competition for fluency you can set a can in circle 9 and have the students drive out to the can at a calculated mav () function speed having them adjust the teacher provided msleep() argument with addition or subtraction equations within the argument to successfully touch the can at different speeds.

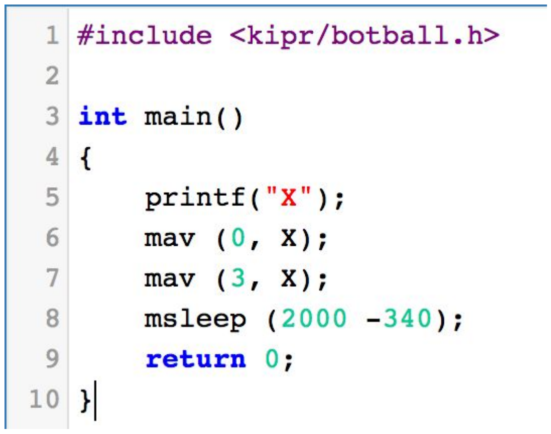
Refer to the two examples of code below.



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

int main()
{
    printf("X");
    mav (0, X);
    mav (3, X);
    msleep (2000 + 850);
    return 0;
}
```

Adding time to the msleep



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

int main()
{
    printf("X");
    mav (0, X);
    mav (3, X);
    msleep (2000 -340);
    return 0;
}
```

Subtracting time from the msleep

Teacher will give the speed will be 200 + 600+ 300 (Remember the mav () function goes from -1500 to 1500) The students will calculate the speed load it into the program and run it to try and touch the can in circle 9. They must then adjust the msleep value of 2000 by adding to it or subtracting time until they successfully touch can 9.

Goals

- In a competitive environment, students will practice their fluency in adding and subtracting numbers within the mav function argument

Outcomes

- The students will solve with fluency one-step and two-step problems involving addition and subtraction within 1,000 using strategies based on place value, properties of operations, and the relationship between addition and subtraction
- The students will be able to represent one- and two-step problems involving addition and subtraction of whole numbers to 1,000 using equations