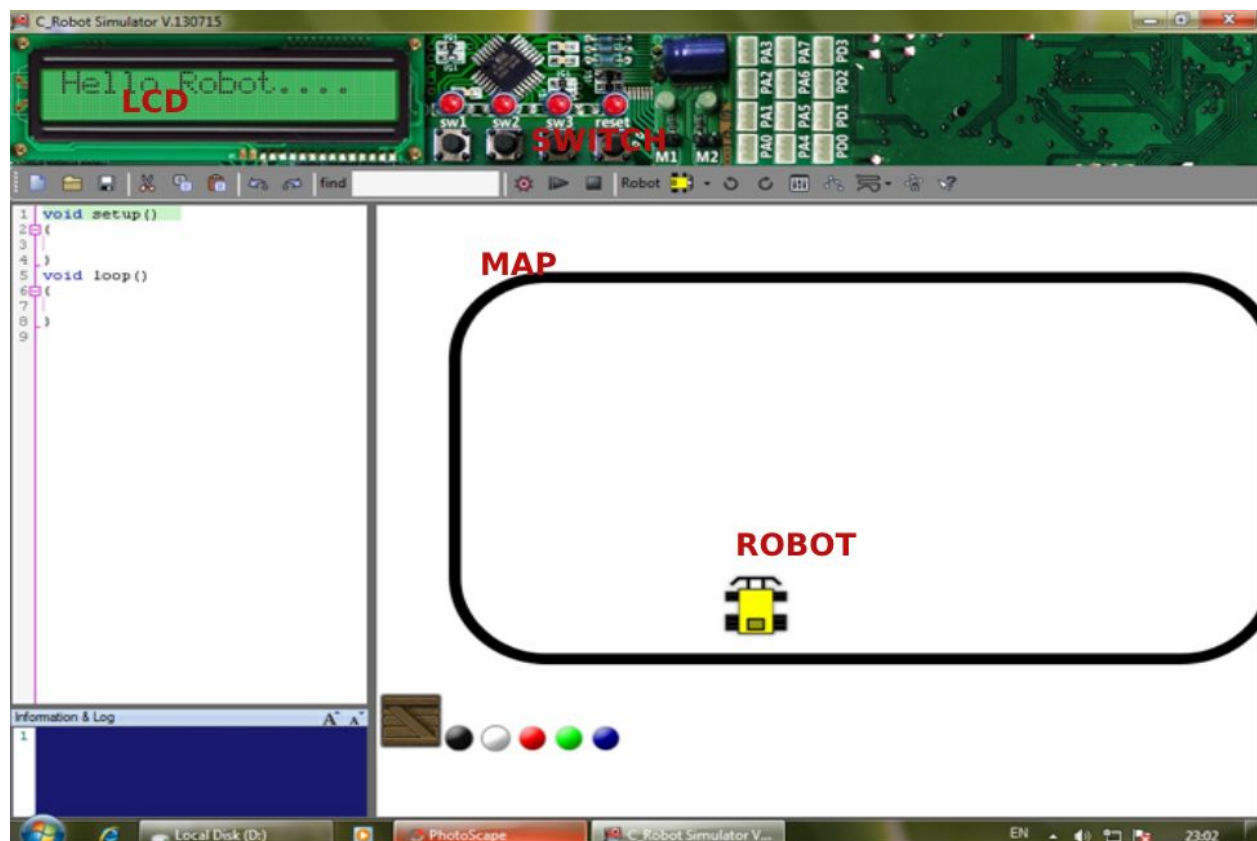# Simulator API

Using C/C++ Simulator

After installing and opening the simulator you will see a robot on a square map on the right and a space to write your codes on the left. The right screen is the environment of the robot and the left screen is your code screen. The robot on the right will behave according to the code written on the left. The code/program screen has two sub-programs: void setup() and void loop(). The statements on the void setup are run only once while the statements on the void loop sub programs keep looping until stopped by the user or and end condition is reached. The simulator also provides one LCD screen for display and four switches on the top portion.

The simulator provides a range of functions for moving and controlling the robot. You can see the detailed description here. For now, we are going to discuss the basic movement functions only.

Basic Movement of the Robot

The following functions control the movement of the robot in the simulator:

FUNCTION	TASK
fd(int p)	Move the robot forward with power of p in both the wheels.
bk(int p)	Move the robot backward with power p in both the wheels.
tr(int p)	Turns the robot right. Right wheel remains stationary and the left wheel turns forward with power p. Soft Turn.*
tl(int p)	Turns the robot left. Left wheel remains stationary and the right wheel turns forward with power p. Soft Turn. *
sr(int p)	Spins the robot right. Left wheel turns forward with power p while the right wheel turns backward with power p. Hard Turn. **
sl(int p)	Spins the robot left. Right wheel turns forward with power p while the left wheel turns backward with power p. Hard Turn. **
fd2(int a, int b)	General forward motion. Turns the left wheel forward with power a and the right wheel forward with power b.

bk2(int a, int b)	General backward motion. Turns the left wheel backward with power a and the right wheel backward with power b.
sleep(int t)	Hold the program execution upto the previous line for t milliseconds.

*Soft Turn

A soft turn is the turning motion of the robot in which one wheel remains stationary and the other moves. The robot turns in the direction of the stationary wheel. The center of turning in this case is at the stationary wheel and the turn radius is equal to the distance between the wheels.
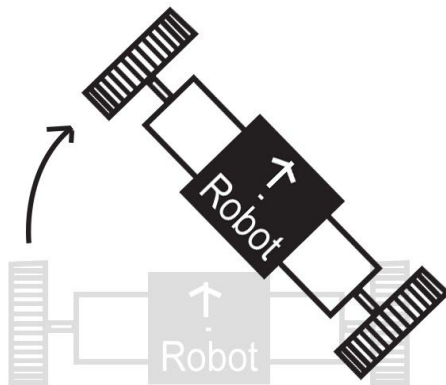

fig:- Soft Turn

**Hard Turn

A hard turn is the spinning motion of the robot in which the wheels of the robot turn in opposite direction with equal velocity. In this case the robot spins in a place like a top. The center of rotation is the center of the line joining the two wheels. The turning radius is equal to the distance from either of the wheels to the center of rotation.
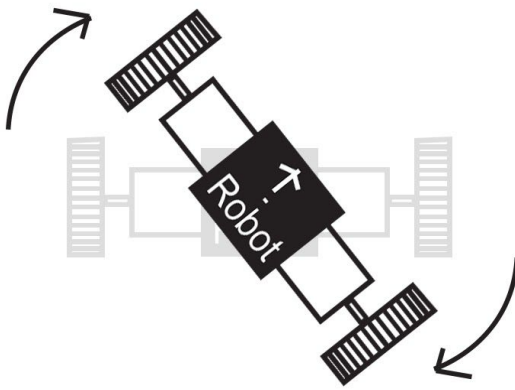
fig:- Hard Turn

Now that you have learned the movements of the robot it is time to try some free movement by yourself. Implement the following code in the simulator and observe the behavior of your robot.

```
//code to move robot forward and stop
void setup()
{

}
void loop()
{
    fd(80);

    sleep(1000);

    system_stop ();
}
```

```
//code to move robot back and stop
void setup()
{

}
void loop()
{
    bk(80);

     sleep(1000);

   system_stop ();
}
```

```
//code to move robot left and stop
void setup()
{

}
void loop()
{
    tl(80);

     sleep(1000);

   system_stop ();
}
```

```
//code to move robot  in circular way
void setup()
{

}
void loop()
{
    fd2(30,60);
}
```

```
//code to move robot  in  square

void setup()
{

}
void loop()
{

  fd(80);
  sleep(1500);
   fd2(0,93);
    sleep(1000);
}
```

Also observe the [following code](#) to run your robot in a square track in the simulator.

Now write your own code and observe the difference between hard turn and soft turn.

When do you think each type of turn is used?