```
//Data type used to tell firing routine the power needed to launch the catapult with
enum Distance{
        SHORT,
        MID,
        TILE,
        CORNER
};


//Data type used to know when the arm is available for firing, and to contrl the arm
//Also tells the arm when to fire
enum ArmStatus{
        ARM_FIRE,
        ARM_LOAD
};


//Data type used to tell the gate to open or close
enum GateStatus{
        GATE_OPEN,
        GATE_CLOSE
};


//Sets arm and gate quadrature encoders to zero
void resetSensors();


/*
        Sets a specific side of the robot to drive forward or backwards

        bool leftSide:
                If true, function sets left side drive motors and NOT right side
                If false, function sets right side drive motors and NOT left side
        bool backwards:
                If true, the speed given is negated, and the specified side drive backwards
                If false, robot drives forward
        int speed:
                The power of the motors ranging from 0 to 127
*/
void drive(bool leftSide, bool backwards, int speed);


/*
```

```
        Configures drive motors to rotate slowly at desired speed and direction for aiming
        Uses ROTATE_POWER for speed

        bool left:
                If true, robot rotates counter-clockwise
                If false, robot rotates clockwise
*/
void rotate(bool left);


/*
        Configures drive motors to drive sideways in desired direction
        Uses MECHANUM_POWER for speed

        bool left:
                If true, robot strafes to the left
                If false, robot strafes to the right
*/
void mechanumDrive(bool left);


//Sets drive motors based on analog stick channels 2 and 3
void tankDrive();


//Stops left side drive motors
void stopLeftDrive();


//Stops right side drive motors
void stopRightDrive();


//Stops all drive motors
void stopAllDrive();


//Returns arm speed based on current distance variable
int getArmSpeed();


/*
        Set distance for next launch
```

```
        Distance newDistance:
                An enumerated type used to define four launch distances
                SHORT, MID, TILE, CORNER
*/
void setDistance(Distance newDistance);


/*
        Sets the arm status

        ArmStatus status:
                An enumerated type used to define two states of arm operation
                ARM_FIRE, ARM_LOAD
*/
void setArm(ArmStatus status);


/*
        Sets arm motors to desired speed and direction

        bool load:
                If true, arm is driven backwards at given speed
                If false, arm is fired at given speed
        int speed:
                The power of the motors ranging from 0 to 127
*/
void fire(bool load, int speed);


//Stops arm motors
void stopArm();


/*
        Sets the gate status

        GateStatus status:
                An enumerated type used to define two states of gate location
                GATE_CLOSE, GATE_OPEN
*/
void setGate(GateStatus status);


//Monitors arm position, and moves arm when prompted by armStatus variable
```

```
task armPosition();


//Monitors gate position, and moves arm when prompted by gateStatus variable
task gatePosition();
```