```
#include "Main.h"
void main ( void )
{
    int startButton;
    int startButtonPortNumber;
    int backupCounter = 0;
    int MAX_BACKUP_WAIT_COUNTER = 1000;
    freq = 0; // 0=1khz (red), 1=10kHz(green beacon)
    ambient_level = 45; // // was 100 // was 200 //used in 'move'
    slow_level = 5000; // used in 'move'
    stop_level = 6000; // used in 'move'
    expose_time = 3; // used in expose_and_read
    steer_sensitivity = 20; // used in 'move'
    forward_speed = 50; // forward speed, used in 'move' (was 35)
    slow_speed = 25; // slow speed, used in 'move'
    spin_speed = 50; // spin speed (for searching mode), used in 'move'
    SetDigitalOutput ( 10 , freq ) ; // turn to 1kHz (red beacon) ...change 12 to 10???
    leftMotorPortNumber = 2; // Set the Left Motor to Motor Port #2
    rightMotorPortNumber = 9; // Set the Right Motor to Motor Port #9
    limitSwitchPortNumber = 1; // The Analog (Digital?) Port number the limit switch is connected to.
    isRedBeaconFound = FALSE;
    armServoPortNumber = 3;
    leftBackBumperPortNumber = 2; // was 5
    rightBackBumperPortNumber = 4; // was 6
    leftBackBumper = NOT_PRESSED;
    rightBackBumper = NOT_PRESSED;
    Wait ( 1000 ) ;
    SetServo ( armServoPortNumber , -127 ) ;
    startButtonPortNumber = 3;
    startButton = NOT_PRESSED ;
    // Wait until start button is pressed:
    PrintToScreen ( "startButton = %d\n" , startButton ) ;
    while ( startButton == NOT_PRESSED  )
    {
        startButton = GetDigitalInput ( startButtonPortNumber ) ;
        PrintToScreen ( "startButton = %d\n" , startButton ) ;
    }
    // Find Red Beacon:
    while ( isRedBeaconFound == FALSE  )
    {
        Read_PD (   ) ;
        find_max ( ) ;
        isRedBeaconFound = move ( ) ;
    }
    SetMotor ( rightMotorPortNumber , 0 ) ;
    SetMotor ( leftMotorPortNumber , 0 ) ;
    // hit switch until switch is off:
    isRedBeaconOn = TRUE;
    while ( isRedBeaconOn  )
    {
        hitSwitch ( ) ;
        // check if red beacon still on:
        Wait ( 2000 ) ;
        Read_PD ( ) ;
        if ( PD_sum <= ambient_level  )
```

```
        {
            isRedBeaconOn = FALSE;
        }
    }
    // backup:
    SetMotor ( rightMotorPortNumber , 100 ) ;
    SetMotor ( leftMotorPortNumber , -100 ) ;
    Wait ( 1000 ) ;
    SetMotor ( leftMotorPortNumber , 0 ) ;
    SetMotor ( rightMotorPortNumber , 0 ) ;
    // Now go for green beacon:
    isGreenBeaconFound = FALSE;
    freq = 1; // DEBUG: testing red again! // 0=1khz (red), 1=10kHz(green beacon)
    SetDigitalOutput ( 10 , freq ) ; // turn to 1kHz (red beacon) ...change 12 to 10???
    while ( isGreenBeaconFound  == FALSE )
    {
        Read_PD (   ) ;
        find_max ( ) ;
        isGreenBeaconFound = move ( ) ;
    }
    SetMotor ( rightMotorPortNumber , 0 ) ;
    SetMotor ( leftMotorPortNumber , 0 ) ;
    // Capture Green Beacon:
    // put arm up then down
    SetServo ( armServoPortNumber , -127 ) ; // put arm up
    Wait ( 1000 ) ;
    SetServo ( armServoPortNumber , -30 ) ; // put arm down
    Wait ( 1000 ) ;
    // backup:
    SetMotor ( rightMotorPortNumber , 80 ) ;
    SetMotor ( leftMotorPortNumber , -80 ) ;
    Wait ( 500 ) ;
    backupCounter = 0;
    //MAX_BACKUP_WAIT_COUNTER = 10000000;
    leftBackBumper = NOT_PRESSED;
    rightBackBumper = NOT_PRESSED;
    // [!] ISSUE: only backs up for 500 ms then stops instantly....
    startButton = NOT_PRESSED;
    while ( leftBackBumper == NOT_PRESSED && rightBackBumper == NOT_PRESSED && startButton == NOT_¬
    PRESSED  ) // backupCounter < MAX_BACKUP_WAIT_COUNTER
    {
        //backupCounter = backupCounter + 1;
        // Read Back Bumpers
        leftBackBumper = GetDigitalInput ( leftBackBumperPortNumber ) ;
        rightBackBumper = GetDigitalInput ( rightBackBumperPortNumber ) ;
        if ( leftBackBumper == PRESSED  || rightBackBumper == PRESSED )
        {
            //backupCounter = MAX_BACKUP_WAIT_COUNTER ;
            // stop both motors:
            SetMotor ( leftMotorPortNumber , 0 ) ;
            SetMotor ( rightMotorPortNumber , 0 ) ;
            // go forward a little:
            SetMotor ( leftMotorPortNumber , 80 ) ;
            SetMotor ( rightMotorPortNumber , -80 ) ;
            Wait ( 500 ) ;
```

```
            SetMotor ( leftMotorPortNumber , 0 ) ;
            SetMotor ( rightMotorPortNumber , 0 ) ;
            // turn a little:
            SetMotor ( leftMotorPortNumber , -80 ) ;
            SetMotor ( rightMotorPortNumber , -80 ) ;
            Wait ( 600 ) ;
            SetMotor ( leftMotorPortNumber , 0 ) ;
            SetMotor ( rightMotorPortNumber , 0 ) ;
            // contimue going back:
            // backup:
            SetMotor ( rightMotorPortNumber , 80 ) ;
            SetMotor ( leftMotorPortNumber , -80 ) ;
            // reset bumpers as not pressed:
            leftBackBumper = NOT_PRESSED;
            rightBackBumper = NOT_PRESSED;
        }
        startButton = GetDigitalInput ( startButtonPortNumber ) ;
    }
    // stop both motors:
    SetMotor ( leftMotorPortNumber , 0 ) ;
    SetMotor ( rightMotorPortNumber , 0 ) ;
}
```