

```

// Include necessary libraries
#include<Adafruit_LiquidCrystal.h>
#include <Keypad.h>
#include <Stepper.h>

// Define pin numbers
int potPinButton = 2;           // Voltage sensor for resistor/button
array (wired to Analog 2)
int potPinSwitch = 3;          // Voltage sensor for resistor/switch
array (wired to Analog 3)
int opto = 9;                   // Optocoupler for triggering camera
(wired to Digital 9)

// Define and initialize variables
const byte rows = 4;           // Number of keypad rows
const byte cols = 3;           // Number of keypad columns
int rowCount = 1;              // Row counter for menus
int menuCount = 1;             // Menu counter
int page = 1;                  // Submenu counter
int potValButton = 0;          // Value of potPinButton
int potValSwitch = 0;          // Value of potPinSwitch
long d = 0;                     // Distance slider moves along the rail (in
motor steps)

char timeDigit1 = 95;          // Hundreds digit of run time (initially set
to display "_")
char timeDigit2 = 95;          // Tens digit of run time (initially set to
display "_")
char timeDigit3 = 95;          // Ones digit of run time (initially set to
display "_")
char exposureDigit1 = 95;      // Tens digit of exposure (initially set to
display "_")
char exposureDigit2 = 95;      // Ones digit of exposure (initially set to
display "_")
char exposureDigit3 = 95;      // Tenths digit of exposure time (initially
set to display "_")
char exposureDigit4 = 95;      // Hundredths digit of exposure time
(initially set to display "_")
char exposureDigit5 = 95;      // Thousands digit of exposure time
(initially set to display "_")
char periodDigit1 = 95;        // Tens digit of period (initially set to
display "_")

```



```

1...
    page = 2;                // Go to page 2
    }
    else{                    // If user is currently on page
2...
    page = 1;                // Go to page 1
    }
    updateMenu();           // Update and display the menu
    }
}

void down() {                // Function for moving the cursor down
one row in the menu
    delay(200);              // Wait 0.2 sec for button bounce to
settle
    if(rowCount < 3) {      // If the user is in row 1 or 2...
        rowCount = rowCount + 1; // Go down one row
        updateMenu();        // Update and display the menu
    }
    else{                   // If the user goes down from row
3...
        rowCount = 1;        // Go to row 1
        if(page == 1){       // If user is currently on page
1...
            page = 2;        // Go to page 2
            }
            else{            // If user is currently on page
2...
                page = 1;    // Go to page 1
            }
            updateMenu();    // Update and display the menu
        }
    }

void select() {              // Function for selecting menu options
    delay(200);              // Wait 0.2 sec for button bounce to
settle
    switch(menuCount) {      // Check which menu the user is on
        case 1:              // If the user is on the Main
Menu...
            switch(page) {   // Check which page the user
is on

```

```

        case 1:                                // If the user is on
Page 1...
        switch(rowCount){                    // Check which row
the user is on
        case 1:                                // If the user
selected row 1 (Calibrate)...
            menuCount = 3;                    // Set menu
index to 3 (calibrate menu)
            rowCount = 1;                    // Reset row
counter to 1
            lcd.clear();                    // Clear LCD
            updateMenu();                    // Update
and display the menu
            break;
        case 2:                                // If the user
selected row 2 (Reset)...
            menuCount = 2;                    // Set menu
index to 2 (reset menu)
            rowCount = 1;                    // Reset row
counter to 1
            lcd.clear();                    // Clear LCD
            updateMenu();                    // Update
and display the menu
            break;
        case 3:                                // If the user
selected row 3 (Timelapse)...
            menuCount = 4;                    // Set menu
index to 4 (timelapse menu)
            rowCount = 1;                    // Reset row
counter to 1
            lcd.clear();                    // Clear LCD
            updateMenu();                    // Update
and display the menu
            break;
        }
        break;
        case 2:                                // If the user is on
Page 2...
        switch(rowCount){                    // Check which row
the user is on
        case 1:                                // If the user
selected row 1 (Video)...

```

```

        menuCount = 5; // Set menu
index to 5 (video menu)
        rowCount = 1; // Reset row
counter to 1
        lcd.clear(); // Clear LCD
        updateMenu(); // Update
and display the menu
        break;
        case 2: // If the user
selected row 3 (Battery)...
        batteryCheck(); // Check the
battery voltage
        break;
        case 3: // If the user
selected row 2 (Settings)...
        menuCount = 6; // Set menu
index to 6 (settings menu)
        rowCount = 1; // Reset row
counter to 1
        lcd.clear(); // Clear LCD
        updateMenu(); // Update
and display the menu
        break;
    }
    break;
}
break;
case 2: // If the user is on the Reset
Menu...
    switch(rowCount){ // Check which row the user
is on
        case 1: // If the user selected
row 1 (Left)...
            left(); // Move the slider
left
            break;
        case 2: // If the user selected
row 2 (Right)...
            right(); // Move the slider
right
            break;
        case 3: // If the user selected

```

```

row 3 (Back)...
    menuCount = 1; // Set menu index
back to 1 (main menu)
    page = 1; // Set page count
back to 1 (to display page 1 of main menu)
    rowCount = 2; // Reset row counter
back to 2
    lcd.clear(); // Clear the LCD
    updateMenu(); // Update and
display the menu
    break;
}
break;
case 3: // If the user is on the
Calibrate Menu...
    switch(rowCount){ // Check which row the user
is on
        case 1: // If the user selected
row 1 (Start)...
            calibrate(); // Run the
calibration
            break;
        case 3: // If the user selected
row 3 (Back)...
            menuCount = 1; // Set menu index
back to 1 (main menu)
            page = 1; // Set page count
back to 1 (to display page 1 of the main menu)
            rowCount = 1; // Reset row counter
back to 1
            lcd.clear(); // Clear the LCD
            updateMenu(); // Update and
display the menu
            break;
        }
    break;
case 4: // If the user is on the
Timelapse Menu...
    switch(page){ // Check which page the user
is on
        case 1: // If the user is on
Page 1...

```

```

        switch(rowCount){ // Check which row
the user is on
        case 1: // If the user
selected row 1 (Run Time)...
            lcd.setCursor(13,1); // Set the
cursor to the first digit
            lcd.blink(); // Set the
cursor to flash
            timeInput(); // Allow
user to input values for run time
            break;
        case 2: // If the user
selected row 2 (Period Time)...
            lcd.setCursor(12,2); // Set the
cursor to the first digit
            lcd.blink(); // Set the
cursor to flash
            periodInput(); // Allow
user to input values for period time
            break;
        case 3: // If the user
selected row 3 (Exposure Time)...
            lcd.setCursor(12,3); // Set the
cursor to the first digit
            lcd.blink(); // Set the
cursor to flash
            exposureInput(); // Allow
user to input values for exposure time
            break;
        }
        break;
        case 2: // If the user is on
Page 2...
            switch(rowCount){ // Check which row
the user is on
        case 1: // If the user
selected row 1 (Run Left)...
            timelapseLeft(); // Run the
timelapse to the left
            break;
        case 2: // If the user
selected row 2 (Run Right)...

```



```

        timelapseRight(); // Run the
timelapse to the right
        break;
        case 3: // If the user
selected row 3 (Back)...
            menuCount = 1; // Set menu
index back to 1 (main menu)
            page = 1; // Set page
count back to 1 (to display page 1 of the main menu)
            rowCount = 3; // Reset row
counter back to 3
            lcd.clear(); // Clear the
LCD
            updateMenu(); // Update
and display the menu
            break;
        }
        break;
    }
    break;
    case 5: // If the user is on the Video
Menu...
        switch(rowCount) { // Check which row the user
is on
            case 1: // If the user selected
row 1 (Set Speed)...
                lcd.setCursor(13,1); // Set the cursor to
the first digit
                lcd.blink(); // Set the cursor to
flash
                speedInput(); // Allow user to
input values for motor speed
                break;
            case 2: // If the user selected
row 2 (Run)...
                menuCount = 7; // Set menu index to
7 (video run menu)
                rowCount = 1; // Reset row counter
back to 1
                lcd.clear(); // Clear the LCD
                updateMenu(); // Update and
display the menu

```

```

        break;
        case 3: // If the user selected
row 3 (Back)...
            menuCount = 1; // Set menu index
back to 1 (main menu)
            page = 2; // Set page count
back to 2 (to display page 2 of the main menu)
            rowCount = 1; // Reset row counter
back to 1
            lcd.clear(); // Clear the LCD
            updateMenu(); // Update and
display the menu
            break;
        }
        break;
        case 6: // If the user is on the
Settings Menu...
            switch(rowCount) { // Check which row the user
is on
                case 1: // If the user selected
row 1 (Backlight On)...
                    lcd.setBacklight(HIGH); // Turn the LCD
backlight on
                    break;
                case 2: // If the user selected
row 2 (Backlight Off)...
                    lcd.setBacklight(LOW); // Turn the LCD
backlight off
                    break;
                case 3: // If the user selected
row 3 (Back)...
                    menuCount = 1; // Set menu index
back to 1 (main menu)
                    page = 2; // Set page count
back to 2 (to display page 2 of the main menu)
                    rowCount = 3; // Reset row counter
back to 3
                    lcd.clear(); // Clear the LCD
                    updateMenu(); // Update and
display the menu
                    break;
            }
    }

```

```

    break;
    case 7: // If the user is on the Run
Video Menu...
        switch(rowCount) { // Check which row the user
is on
            case 1: // If the user selected
row 1 (Left)...
                runVideoLeft(); // Run the video left
                break;
            case 2: // If the user selected
row 2 (Right)...
                runVideoRight(); // Run the video
right
                break;
            case 3: // If the user selected
row 3 (Back)...
                menuCount = 5; // Set the menu
index back to 5 (video menu)
                rowCount = 2; // Reset row counter
back to 2
                lcd.clear(); // Clear the LCD
                updateMenu(); // Update and
display the menu
                break;
        }
    }
}

```

// Function to update menu

```

void updateMenu() {
    switch(menuCount) { // Check the value of the menu counter
        case 1: // If the menu counter is 1..
            switch(page) { // Check the value of the page
counter
                case 1: // If the page counter is
1...
                    mainMenu1(); // Display page 1 of
the main menu
                    break;
                case 2: // If the page counter is
2...

```

```

        mainMenu2(); // Display page 2 of
the main menu
        break;
    }
break;
case 2: // If the menu counter is 2...
    resetMenu(); // Display the reset menu
break;
case 3: // If the menu counter is 3...
    calibrateMenu(); // Display the calibrate menu
break;
case 4: // If the mneu counter is 4...
    switch(page){ // Check the value of the page
counter
        case 1: // If the page counter is
1...
            runMenu1(); // Display page 1 of
the timelapse menu
            break;
        case 2: // If the page counter is
2...
            runMenu2(); // Display page 2 of
the timelapse menu
            break;
    }
break;
case 5: // If the menu counter is 5...
    videoMenu(); // Display the video menu
break;
case 6: // If the menu counter is 6...
    settingsMenu(); // Display the settings menu
break;
case 7: // If the menu counter is 7...
    videoRunMenu(); // Display the run menu for
video mode
    break;
}
}

```

```

// Functions for inputing values for motor speed, exposure time, total
time, and period using keypad

```



```

    }
}
lcd.noBlink(); // Stop the cursor from
flashing
}

void exposureInput() { // Function for setting exposure
time with keypad
    int digit = 1; // Create digit counter and
initialize to 1 (tens place)
    while(digit < 6){ // While digit counter is
less than 6...
        char key = keypad.getKey(); // Retrieve status from
keypad
        if(key != NO_KEY){ // If the keypad is
pressed...
            delay(250); // Wait a 0.25
for button bounce to settle
            switch(digit){ // Check which
digit the user is on
                case 1: // If the
user is entering digit 1 (tens place)...
                    exposureDigit1 = key; //
Save the value of the pressed key to digit 1 of exposure
                    lcd.print(key); //
Display the pressed key on the LCD
                    digit = digit + 1; //
Move to the next digit (ones place)
                    break;
                case 2: // If the
user is entering digit 2 (ones place)...
                    exposureDigit2 = key; //
Save the value of the pressed key to digit 2 of exposure
                    lcd.print(key); //
Display the pressed key on the LCD
                    digit = digit + 1; //
Move to the next digit (tenths place)
                    lcd.setCursor(15,3); //
Move the cursor to the tenths place (skip decimal point)
                    break;
                case 3: // If the
user is entering digit 3 (tenths place)...

```

```

        exposureDigit3 = key;                                //
Save the value of the pressed key to digit 3 of exposure
        lcd.print(key);                                    //
Display the pressed key on the LCD
        digit = digit + 1;                                //
Move to the next digit (hundredths place)
        break;
        case 4:                                           // If the
user is entering digit 4 (hundredths place)...
        exposureDigit4 = key;                                //
Save the value of the pressed key to digit 4 of exposure
        lcd.print(key);                                    //
Display the pressed key on the LCD
        digit = digit + 1;                                //
Move to the next digit (thousandths place)
        break;
        case 5:                                           // If the
user is entering digit 5 (thousandths place)...
        exposureDigit5 = key;                                //
Save the value of the pressed key to digit 5 of exposure
        lcd.print(key);                                    //
Display the pressed key on the LCD
        digit = digit + 1;                                //
Move to the next digit (exit while loop)
        break;
    }
}
}
    lcd.noBlink();                                        // Stop the cursor from
flashing
}

void timeInput(){
    int digit = 1;
    while(digit < 4){
        char key = keypad.getKey();
        if(key != NO_KEY){
            delay(250);
            switch(digit){
                case 1:
                    timeDigit1 = key;
                    lcd.print(key);

```

```

        digit = digit + 1;
        break;
    case 2:
        timeDigit2 = key;
        lcd.print(key);
        digit = digit + 1;
        break;
    case 3:
        timeDigit3 = key;
        lcd.print(key);
        digit = digit + 1;
        break;
    }
}
}
}
lcd.noBlink();
}

```

```

void periodInput() {
    int digit = 1;
    while(digit < 4){
        char key = keypad.getKey();
        if(key != NO_KEY){
            delay(250);
            switch(digit){
                case 1:
                    periodDigit1 = key;
                    lcd.print(key);
                    digit = digit + 1;
                    break;
                case 2:
                    periodDigit2 = key;
                    lcd.print(key);
                    digit = digit + 1;
                    lcd.setCursor(15,2);
                    break;
                case 3:
                    periodDigit3 = key;
                    lcd.print(key);
                    digit = digit + 1;
                    break;
            }
        }
    }
}

```



```

    }
}
lcd.noBlink();
}

// Functions to display menus

void mainMenu1() { // Page 1 of Main Menu
(Menu 1, Page 1)
    lcd.setCursor(0,0); // Set cursor to
Column 0, Row 0 (top left corner of LCD)
    lcd.print("Main Menu - Page 1:"); // Display "Main
Menu - Page 1" on LCD
    lcd.setCursor(0,1); // Set cursor to
Column 0, Row 1 (second row of LCD)
    lcd.print("  Calibrate          "); // Display
"Calibrate" on LCD, leaving room for arrow on left side
    lcd.setCursor(0,2); // Set cursor to
Column 0, Row 2 (third row of LCD)
    lcd.print("  Reset              "); // Display "Reset"
on LCD, leaving room for arrow on left side
    lcd.setCursor(0,3); // Set cursor to
Column 0, Row 3 (fourth row of LCD)
    lcd.print("  Timelapse          "); // Display
"Timelapse" on LCD, leaving room for arrow on left side
    lcd.setCursor(0,rowCount); // Set the cursor to
the current row that the user is on
    lcd.print("->"); // Display an arrow
on LCD
}

void mainMenu2() { // Page 2 of Main Menu
(Menu 1, Page 2)
    lcd.setCursor(0,0);
    lcd.print("Main Menu - Page 2:");
    lcd.setCursor(0,1);
    lcd.print("  Video              ");
    lcd.setCursor(0,2);
    lcd.print("  Battery: ");
    lcd.print(voltage);
    lcd.print("%    ");
    lcd.setCursor(0,3);

```

```
    lcd.print(" Settings      ");
    lcd.setCursor(0,rowCount);
    lcd.print("->");
}
```

```
void calibrateMenu() { // Calibrate Menu (Menu
2)
    lcd.setCursor(0,0);
    lcd.print("Calibrate:");
    lcd.setCursor(0,1);
    lcd.print(" Start");
    lcd.setCursor(0,2);
    lcd.print(" Distance: ");
    lcd.print(d);
    lcd.setCursor(0,3);
    lcd.print(" Back");
    lcd.setCursor(0,rowCount);
    lcd.print("->");
}
```

```
void resetMenu() { // Reset Menu (Menu 3)
    lcd.setCursor(0,0);
    lcd.print("Reset:");
    lcd.setCursor(0,1);
    lcd.print(" Left");
    lcd.setCursor(0,2);
    lcd.print(" Right");
    lcd.setCursor(0,3);
    lcd.print(" Back");
    lcd.setCursor(0,rowCount);
    lcd.print("->");
}
```

```
void runMenu1() { // Page 1 of Timelapse
Menu (Menu 4, Page 1)
    lcd.setCursor(0,0);
    lcd.print("Timelapse - Page 1:");
    lcd.setCursor(0,1);
    lcd.print(" Run Time: ");
    lcd.print(timeDigit1);
    lcd.print(timeDigit2);
    lcd.print(timeDigit3);
```

```

    lcd.print(" min");
    lcd.setCursor(0,2);
    lcd.print("  Period:  ");
    lcd.print(periodDigit1);
    lcd.print(periodDigit2);
    lcd.print(".");
    lcd.print(periodDigit3);
    lcd.print(" sec");
    lcd.setCursor(0,3);
    lcd.print("  Exposure: ");
    lcd.print(exposureDigit1);
    lcd.print(exposureDigit2);
    lcd.print(".");
    lcd.print(exposureDigit3);
    lcd.print(exposureDigit4);
    lcd.print(exposureDigit5);
    lcd.print(" s");
    lcd.setCursor(0,rowCount);
    lcd.print("->");
}

```

```

void runMenu2() { // Page 2 of Timelapse
Menu (Menu 4, Page 2)
    lcd.setCursor(0,0);
    lcd.print("Timelapse - Page 2:");
    lcd.setCursor(0,1);
    lcd.print("  Run Left          ");
    lcd.setCursor(0,2);
    lcd.print("  Run Right          ");
    lcd.setCursor(0,3);
    lcd.print("  Back              ");
    lcd.setCursor(0,rowCount);
    lcd.print("->");
}

```

```

void videoMenu() { // Video Menu (Menu 5)
    lcd.setCursor(0,0);
    lcd.print("Video:");
    lcd.setCursor(0,1);
    lcd.print("  Set Speed: ");
    lcd.print(speedDigit1);
    lcd.print(speedDigit2);
}

```

```

lcd.print(speedDigit3);
lcd.print(" rpm");
lcd.setCursor(0,2);
lcd.print("  Run          ");
lcd.setCursor(0,3);
lcd.print("  Back          ");
lcd.setCursor(0,rowCount);
lcd.print("->");
}

void settingsMenu() { // Settings Menu (Menu 6)
  lcd.setCursor(0,0);
  lcd.print("Settings:");
  lcd.setCursor(0,1);
  lcd.print("  Backlight On");
  lcd.setCursor(0,2);
  lcd.print("  Backlight Off");
  lcd.setCursor(0,3);
  lcd.print("  Back");
  lcd.setCursor(0, rowCount);
  lcd.print("->");
}

void videoRunMenu() { // Run Video Menu (Menu
7)
  lcd.setCursor(0,0);
  lcd.print("Run:");
  lcd.setCursor(0,1);
  lcd.print("  Left");
  lcd.setCursor(0,2);
  lcd.print("  Right");
  lcd.setCursor(0,3);
  lcd.print("  Back");
  lcd.setCursor(0, rowCount);
  lcd.print("->");
}

// Programs to move the motor (Reset, Video, and Timelapse modes)

void right() { //
Function to move cart continuously right along the rail until left limit
switch contacts angle bracket

```

```

    myStepper.setSpeed(150); // Set
motor speed to 150 rpm
    int potValSwitch = analogRead(potPinSwitch); //
Check voltage of switch/resistor array
    while(potValSwitch < 480 || potValSwitch > 540){ //
While the voltage is not 2.5 V (potVal = 512) {while left switch is not
pressed}...
    myStepper.
step(-1); // Move
motor 1 step counterclockwise
    potValSwitch =
analogRead(potPinSwitch); // Recheck
voltage of switch/resistor array
    }
}

void left() { //
Function to move cart continuously left along the rail until left limit
switch contacts angle bracket
    myStepper.setSpeed(150); // Set
motor speed to 150 rpm
    int potValSwitch = analogRead(potPinSwitch); //
Check voltage of switch/resistor array
    while(potValSwitch < 310 || potValSwitch > 370){ //
While the voltage is not 1.67 V (potVal = 341) {while right switch is not
pressed}...
    myStepper.
step(1); // Move
motor 1 step clockwise
    potValSwitch =
analogRead(potPinSwitch); // Recheck
voltage of switch/resistor array
    }
}

void calibrate() { //
Function to move cart continuously along the rail from left to right angle
bracket and record number of motor steps
    left(); //
Move cart to the left angle bracket
    delay(2000); //
Wait 2 sec

```

```

    d = 0;
    int potValSwitch = analogRead(potPinSwitch); //
Check voltage of switch/resistor array
    while(potValSwitch < 480 || potValSwitch > 540){ //
While the voltage is not 1.67 V (potVal = 341) {while right switch is not
pressed}...
    myStepper.
step(-1); // Move
motor 1 step clockwise
    d = d +
1; // Add 1
to the motor step count
    potValSwitch =
analogRead(potPinSwitch); // Recheck
voltage of switch/resistor array
}
    updateMenu(); //
Update and display the menu to show the motor step count
    Serial.println(00010000); //
Send signal serially to PIC to sound buzzer
}

void runVideoRight()
{
    // Function to move cart continuously along the rail from left to right
at user selected speed
    long motorSpeed = (speedDigit1 - '0')*100 + (speedDigit2 - '0')*10 +
(speedDigit3 - '0'); // Convert user input for each digit into a
number for motor speed
    myStepper.
setSpeed(motorSpeed);
        // Set the speed of the motor to the value the user inputed
    int potValSwitch =
analogRead(potPinSwitch);
        // Check the voltage value from the switch/resistor array
    while(potValSwitch < 480 || potValSwitch >
540){ // While the
right switch is not pressed (potVal = 512)
    myStepper.
step(-1);
        // Move motor one step to the right
    potValSwitch =

```

```

analogRead(potPinSwitch);
    // Recheck voltage of switch/resistor array
}
}

void runVideoLeft()
{
    // Function to move cart continuously along the rail from right to left
    at user selected speed
    long motorSpeed = (speedDigit1 - '0')*100 + (speedDigit2 - '0')*10 +
(speedDigit3 - '0');
    myStepper.setSpeed(motorSpeed);
    int potValSwitch = analogRead(potPinSwitch);
    while(potValSwitch < 310 || potValSwitch > 370){
        myStepper.step(1);
        potValSwitch = analogRead(potPinSwitch);
    }
}

void timelapseRight(){ // Function to create
move-shoot-move movement for a timelapse moving from left to right

    float pauseTime = 250; // Set pause time to 0.25 sec

    // Calculate motor speed and steps per picture
    float totalTime = (timeDigit1 - '0')*100.0 + (timeDigit2 - '0')*10.0 +
(timeDigit3 -
'0');
        // Calculate total run time from digit values entered by user
with keypad
    float periodTime = (periodDigit1 - '0')*10.0 + (periodDigit2 - '0') +
(periodDigit3 - '0')/10.
0;
    // Calculate time between pictures from digit values entered by user with
keypad
    float exposureTime = (exposureDigit1 - '0')*10.0 + (exposureDigit2 -
'0') + (exposureDigit3 - '0')/10.0 + (exposureDigit4 - '0')/100.0 +
(exposureDigit5 - '0')/1000.0; // Calculate exposure time from digit
values entered by user with keypad
    long n =
totalTime*60/periodTime;

```



```

to the picture count
    int potValSwitch = analogRead(potPinSwitch);           // Check the
voltage value from the switch/resistor array
    if (potValSwitch > 480 && potValSwitch < 540){         // If
the right switch is pressed (potVal = 512)...
        break;                                           //
Stop the timelapse
    }
}
    Serial.println(00010000);                             // Signal
the PIC to play a tone on the buzzer
}

void timelapseLeft() {                                     // Function to run a timelapse
to the left (similar code as to the right)

    float pauseTime = 250;

    // Calculate motor speed and steps per picture
    float totalTime = (timeDigit1 - '0')*100.0 + (timeDigit2 - '0')*10.0 +
(timeDigit3 -
'0');
        // Calculate total run time from values entered by user with
keypad
    float periodTime = (periodDigit1 - '0')*10.0 + (periodDigit2 - '0') +
(periodDigit3 - '0')/10.
0;
    // Calculate time between pictures from values entered by user with keypad
    float exposureTime = (exposureDigit1 - '0')*10.0 + (exposureDigit2 -
'0') + (exposureDigit3 - '0')/10.0 + (exposureDigit4 - '0')/100.0 +
(exposureDigit5 - '0')/1000.0; // Calculate exposure time
    long n =
totalTime*60/periodTime;

        // Calculate number of pictures
    float stationaryTime = (n*exposureTime + 3*n*(pauseTime/1000.0))/60.
0;

        // Calculate time for which slider is stationary
    float moveTime = totalTime - stationaryTime;
    long RPM = d/(200L*moveTime);
    long steps = d/n;

```

```

right();
delay(2000);

long i = 0;
float exposureTimeMili = exposureTime*1000;
myStepper.setSpeed(RPM);
while(i <= n){
    myStepper.step(steps);
    delay(pauseTime);
    digitalWrite(opto, HIGH);
    delay(pauseTime);
    digitalWrite(opto, LOW);
    delay(exposureTimeMili);
    delay(pauseTime);
    i = i + 1;
    int potValSwitch = analogRead(potPinSwitch);
    if (potValSwitch > 310 && potValSwitch < 370){
        break;
    }
}
Serial.println(00010000);
}

void batteryCheck() { // Function for checking and
displaying battery voltage
    Serial.println(01000000); // Signal PIC to check
and send battery voltage // Wait 0.01 sec for PIC
    delay(10); // Get and store voltage
to send // Convert value from
    long digitalVal = Serial.read(); // Update menu to
value from PIC
    voltage = digitalVal/256.0 * 100.0;
PIC to a percentage
    updateMenu();
display voltage value
}

```