```
// FILE: xl320-Read0_1b.nxc - Works!
// DATE: 03/25/22 18:53
//       03/26/22 17:13
// AUTH: P.Oh
// DESC: Command single XL-320 to rotate and display count
// VERS: 0.1a: Rotates to position 700 or 200 and displays count as it does so
//       0.1b: Rotate continuously and display count as it does so
// REFS:H:\me729\lesson-H-jacobians\lab\code\xl320-helloServoRead0_1b.nxc
//      H:\me729\lesson-D-denavitHartenberg\lab\code\xl320-helloWheelMode1_0a.nxc

// NOTE: If factory default XL-320 used, then ID is 0x01
//       ID of 0xFE commands any and all XL-320 motors

#include "xl320-defines1_0a.h"   // XL-320 defines from Control Table
#include "xl320-functions1_0d.h" // P.Oh functions written for XL-320
                                 // 1.0b.h contains XL320_servoRead
                                 // 1.0c.h updated the XL320_servoRead
                                 // 1.0d.h has XL320_TorqueEnable an
XL320_ControlMode
#define ID_ALL_MOTORS 0XFE       // 0XFE commands all XL-320 motors
#define ID_MOTOR01    0X01       // Assumes Motor 1 configured with ID = 1

void rotateOneMotorAbsolutely(float angle01) { //------------------
 // Rotates a single Dynamixel XL-320 motor to desired angle
 // Assumes motor count of 512 denotes 0 degrees.  Uses right-hand rule for
 // rotational direction

  float desiredAngle01InDegrees;   // Angle Motor 1 to move to [deg]
  float degreesPerCount;           // Conversion 0.29 [degrees/count]
  float calculatedCount;           // Count equivalent of desired angle [count]
  int   motor01Offset;             // Motor 1's offset [count]
  float theta01InDegrees;          // Motor 1 angle [counts]
  int   theta01InCounts;           // Motor 1 angle [deg]
  string msg01;                    // dummy string to print values to screen

  motor01Offset = 512; // Set Link 1 at 0 deg (i.e. 512 counts)

  // Note 1: Looking into horn from Top, count > 512 is CCW (i.e. +Z axis)
  // and count < 512 is CW (i.e. -Z axis)
  degreesPerCount = 0.29; // [deg/count] found from XL-320 data sheet

  ClearScreen();
  desiredAngle01InDegrees = angle01;
  theta01InCounts = motor01Offset + desiredAngle01InDegrees/degreesPerCount;

  // Format string so displays nicely on Brick screen
  sprintf(msg01, "Goto [%3.1f] " ,desiredAngle01InDegrees);
  TextOut(0, LCD_LINE2, msg01);

  XL320_servo(ID_MOTOR01, theta01InCounts, 200); // motor position at speed 200
  Wait(2000); // wait about 2 seconds before issuing another command
  PlayTone(TONE_B3,50);

}; // end rotateOneMotorAbsolutely function --------------------------------



task main() {

  bool  orangeButtonPushed;      // Detect Brick Center button state
  bool  rightArrowButtonPushed;  // Detect Brick right arrow button state
  bool  leftArrowButtonPushed;   // Detect Brick left arrow button state
  bool  greyButtonPushed;        // Detect Brick Grey/Abort button state
```

```
  unsigned char data[13];        // 13-byte status packet from RS485
  float degreesPerCount;         // Conversion 0.29 [degrees/count]
  int   rawCount;                // raw (unadjusted) count of XL-320
  float theta1InRad;             // angle of joint 1 [rad]
  int   theta1InCounts;          // Motor 1 angle [counts]
  float theta1InDegrees;         // angle of joint 1 [deg]
  int   motor01Offset;           // Motor 1's offset [count]

  degreesPerCount = 0.29; // [deg/count] found from XL-320 data sheet

  UseRS485();
  RS485Enable();
  // Note: First, use Dynamixel Wizard to set XL-320 to desired baud rate
  // Then, use RS485Uart to match this baud rate e.g. 57600
  RS485Uart(HS_BAUD_57600, HS_MODE_8N1); // 57600 baud, 8bit, 1stop, no parity
  Wait(100);

  // (1) Home to XL-320's center position
  TextOut(0, LCD_LINE1, "Homing..." );
  XL320_servo(ID_ALL_MOTORS, 512, 200); // 512 should be center position
  Wait(2000);
  TextOut(0, LCD_LINE2, "Homed..." );
  PlayTone(TONE_E4, 500);

  // Now that XL-320 is home, raw count should be 512
  // Read XL-320 encoder to check
  XL320_servoRead(ID_MOTOR01);
  Wait(20);
  until(RS485DataAvailable());
  RS485Read(data);
  // data[9] = LO and data[10] HI byte contain XL-320 position
  // Thus formulate the position and display as integer
  rawCount = data[9] + (data[10] << 8);
  // ClearScreen();
  ClearLine(LCD_LINE3); // clears the specific line
  TextOut(0, LCD_LINE3, FormatNum("Raw = %4d" , rawCount));
  motor01Offset = rawCount;  // which should be close to 512

  // (2) Turn off Torque enable so that one can freely turn XL320 axle by hand
  XL320_setTorqueEnable(ID_MOTOR01, 0); // 0 = turn OFF torque enable
  Wait(100);

  // (3) Select Wheel Mode
  XL320_controlMode(ID_ALL_MOTORS, 1); // 1 = Wheel Mode; 2 = Joint Mode
  Wait(20);
  TextOut(0, LCD_LINE5, "In Wheel mode" );
  // Prompt user to begin
  TextOut(0, LCD_LINE6, "Start: hit ->" );
  do {
     rightArrowButtonPushed = ButtonPressed(BTNRIGHT, FALSE);
  } while(!rightArrowButtonPushed);

  ClearScreen();
  // Prompt user to begin
  TextOut(0, LCD_LINE1, "<-/->/ORG CW/CCW/QUIT" );

  do {
      rightArrowButtonPushed = ButtonPressed(BTNRIGHT, FALSE);
      if(rightArrowButtonPushed) {
         TextOut(0, LCD_LINE3, "CCW" );
         XL320_servo(ID_ALL_MOTORS, 0, 100);  // Continuous CCW rotation at speed
100
         // Section 2.4.21 says 0-1023 is CCW; 1024-2047 is CW
```

```
        // http://emanual.robotis.com/docs/en/dxl/x/xl320/#moving-speed
        Wait(20);
      }; // end if
      leftArrowButtonPushed = ButtonPressed(BTNLEFT, FALSE);
      if(leftArrowButtonPushed) {
        TextOut(0, LCD_LINE3, "CW " );
        XL320_servo(ID_ALL_MOTORS, 0, 1024 + 100);  // Continuous CCW rotation at
speed 100
        Wait(20);
      }; // end if

     XL320_servoRead(ID_MOTOR01);
     Wait(20);
     until(RS485DataAvailable());
     RS485Read(data);
     // data[9] = LO and data[10] HI byte contain XL-320 position
     // Thus formulate the position and display as integer
     rawCount = data[9] + (data[10] << 8);
     // Looking at XL-320's horn from above, we observe that
     // 512 < raw count < 1023 should be CCW and +theta wrt +Z-axis
     // so, theta1InCounts should increase from zero
     // 0 < raw count < 512 should be CW and -theta wrt +Z-axis
     // so, theta1InCounts should decrease from zero
     theta1InCounts = rawCount - motor01Offset;
     theta1InDegrees = theta1InCounts * degreesPerCount;

     ClearLine(LCD_LINE6); // clears the specific line
     TextOut(0, LCD_LINE6, FormatNum("Raw = %4d" , rawCount));
     TextOut(0, LCD_LINE7, FormatNum("thetaDeg = %3.2f" , theta1InDegrees));
     orangeButtonPushed = ButtonPressed(BTNCENTER, FALSE);
   } while(!orangeButtonPushed); // end of do-while loop

  // User hit ORG button so reset XL320 and exit gracefully
  // Turn XL-320 torque enable ON (ON/OFF = 1/0)
  ClearScreen();
  XL320_setTorqueEnable(ID_ALL_MOTORS, 0);
  Wait(200);
  TextOut(0, LCD_LINE1, "Torque Enable: OFF..." );

  // Return back to Joint Mode
  XL320_controlMode(ID_ALL_MOTORS, 2); // 1 = Wheel Mode; 2 = Joint Mode
  Wait(200);
  ClearScreen();
  TextOut(0, LCD_LINE3, "Joint mode..." );
  TextOut(0, LCD_LINE4, "Homing..." );
  XL320_servo(ID_ALL_MOTORS, 512, 200); // 512 should be center position
  Wait(2000);
  TextOut(0, LCD_LINE6, "Quitting" );
  PlaySound(SOUND_DOWN);

} // end main
```