```
// FILE: xl320-Read0_1a.nxc - Works!
// DATE: 03/25/22 18:14
// 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
// REFS:H:\me729\lesson-H-jacobians\lab\code\xl320-helloServoRead0_1b.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_0c.h" // P.Oh functions written for XL-320
                                 // 1.0b.h contains XL320_servoRead
                                 // 1.0c.h updated the XL320_servoRead

#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
  int   Position;
  float theta1;                  // angle of joint 1 [rad]
  float theta1InDegrees;         // angle of joint 1 [deg]

  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);

  // 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);

  ClearScreen();
  // Prompt user to begin
  TextOut(0, LCD_LINE1, "Start: hit ->" );
  do {
     rightArrowButtonPushed = ButtonPressed(BTNRIGHT, FALSE);
  } while(!rightArrowButtonPushed);

  // First go to home position
  ClearScreen();
  TextOut(0, LCD_LINE2, "Homing..." );
  Wait(2000);
  theta1InDegrees = 0.0;
  rotateOneMotorAbsolutely(theta1InDegrees);
  Wait(2000);
  PlayTone(TONE_E4, 500);

  ClearScreen();
  // Prompt user to begin
  TextOut(0, LCD_LINE1, "Stop: Press GRAY" );
  TextOut(0, LCD_LINE3, "-> Pos: 700" );
  TextOut(0, LCD_LINE4, "<- Pos: 200" );


  while(true) {
     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
     Position = data[9] + (data[10] << 8);
     // ClearScreen();
     ClearLine(LCD_LINE6); // clears the specific line
     TextOut(10, LCD_LINE6, FormatNum("Pos = %4d" , Position));

     if(ButtonPressed(BTNRIGHT, FALSE)) {
         while(ButtonPressed(BTNRIGHT, FALSE)) {
           // Do nothing, but this check flushes any key presses
         };
         XL320_servo(ID_MOTOR01, 700, 200); // rotate to motor position 700, speed
200
         Wait(200);
     }
     else if(ButtonPressed(BTNLEFT, FALSE)) {
         while(ButtonPressed(BTNLEFT, FALSE)) {
           // Do nothing, but this check flushes any key presses
         };
         XL320_servo(ID_MOTOR01, 200, 200);   // counter-rotate to 0 at speed 200;
         Wait(200);
     }; // end if

  }; // end while;

} // end main
```