Minutes 11/18/05 Team1720 Programmers Workshop
AT Building, Rm. 214, 5-7pm

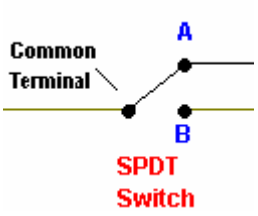
Present: Ashley, Zach, John, Mike

Programming assignments due 11/18:
Ashley: write code to enable both sets of wheel motors on 1020 to be controlled with a correction curve that can override driver input.

Morgan: write code to recognize the tripping of a limit switch to stop movement of the robot arm in one direction, and another limit switch in the other direction.

Obtaining feedback from a sensor and responding to that feedback is fundamental to running a robot in autonomous mode. To understand the most basic sensor input and programmed response, we investigated the use of a limit switch. Mike had mounted a limit switch so that when the robot arm rotated down, the limit switch would trip just before the arm hit it physical limits.

The limit switch was a SPDT (single pole, double throw) switch. The switch had a diagram on the back side showing how the terminals are used. It looks something like this:



In order to recognize switch closure, the normally open pair (Common Terminal and B) are connected between ground and a digital input. (There are three pins on each of the 18 digital ports: ground, +5 and signal in). Digital input pins on the Robot Controller "float" high (+5) if there is no input. When the switch closes (Common Terminal connects to B) as a result of arm rotation, the "signal in" digital input pin is connected to ground causing the previously high signal to go low.

This low signal needs to be used to stop arm rotation so the arm, motor or gears are not damaged. On 1020 the joystick controlling arm rotation is connected to Operator Interface Port 3.

Default_Routine() is used to map Robot Controller PWM outputs to a specific axis of a joystick connected to specific ports: The assignment is:

> pwm09 = p3_y; /* Lift Arm */ pwm output 9 is set to the value of the port 3, Y axis of the joystick.

Pwm09 is connected to the Victor motor controller which is connected to the motor driving the arm.

The limit switch was connected to digital port 9 which was set to be an input in the User_Initialization() function with the following line:

> digital_io_09 = digital_io_10 = digital_io_11 = digital_io_12 = INPUT;

The following function is predefined in user_routines.c:

```
/******************************************************************************
* FUNCTION NAME: Limit_Switch_Max
* PURPOSE:      Sets a PWM value to neutral (127) if it exceeds 127 and the
*               limit switch is on.
* CALLED FROM:  this file
* ARGUMENTS:
*     Argument      Type            IO   Description
*     --------      -------------   --   -----------
```

```
*     switch_state   unsigned char    I    limit switch state
*     *input_value   pointer          O    points to PWM byte value to be limited
* RETURNS:       void
*******************************************************************************/
void Limit_Switch_Max(unsigned char switch_state, unsigned char *input_value)
{
  if (switch_state == CLOSED)
  {
    if(*input_value > 127)
      *input_value = 127;
  }
}
```

The Limit_Swith_Max function has two arguments: switch_state and PulseWidthModulation joystick value. If the switch is closed AND the PWM value is > 127 the PWM value is set to 127.

The limit switch function is called in Default_Routine() (found in user_routines.c) as follows:

```
Limit_Switch_Max(rc_dig_in09, &pwm09);        /* limit switch for arm */
```

This function call evaluates the limit switch state connected to Robot Controller digital input #9 and either limits the output of pwm port 9 to 127 or, if the switch is not closed, allows it to follow the joystick input.

The function was demonstrated by holding the arm joystick in the full “down” position and observation that the arm motor turned off when the limit switch tripped.

A thorough understand of the forgoing demonstration is fundamental to programming the robot and is a prerequisite to any additional programming.