

```

Start Page | Main | Initialize | Autonomous | OperatorControl | Straight_Drive_PID_Loop x
void Straight_Drive_PID_Loop ( long Desired_Distance, int Max_Drive_Speed, unsigned long Max_Time )
{
  PrintToScreen ( "Drive Straight for X distance started\n" );
  PresetTimer ( 1 , 0 );
  PresetQuadEncoder ( 1 , 2 , 0 ); // Left
  PresetQuadEncoder ( 3 , 4 , 0 ); // Right
  if ( Max_Drive_Speed > 127 )
  {
    Max_Drive_Speed = 127 ;
  }
  Max_Drive_Speed = Abs ( Max_Drive_Speed );
  if ( Desired_Distance < 0 )
  {
    Max_Drive_Speed = Max_Drive_Speed * -1 ;
  }
  while ( Drive_Straight_Timer < Max_Time && ( Left_Dist_Error != 0 && Right_Dist_Error != 0 ) )
  {
    Drive_Straight_Timer = GetTimer ( 1 );
    Left_Drive_Encoder = GetQuadEncoder ( 1 , 2 ); // Left
    Right_Drive_Encoder = GetQuadEncoder ( 3 , 4 ); // Right
    // Distance Proportional Loop...
    Left_Dist_Error = Desired_Distance - Left_Drive_Encoder ;
    Right_Dist_Error = Desired_Distance - Right_Drive_Encoder ;
    // Distance Proportional Loop...
    Left_Drive = Left_Dist_Error * Left_Dist_P ;
    Right_Drive = Right_Dist_Error * Right_Dist_P ;
    // Difference PID Loop...
    Diff_Error = ( Left_Drive_Encoder - Right_Drive_Encoder ) ;
    Diff_Integral = ( Diff_Integral + Diff_Error ) ;
    Diff_Deriv = Diff_Error + Prev_Diff_Error ;
  }
}

```

CAP: NUM STM32F103VD | Program size: Unknown | Line: 1 of 60 | 12:47 PM