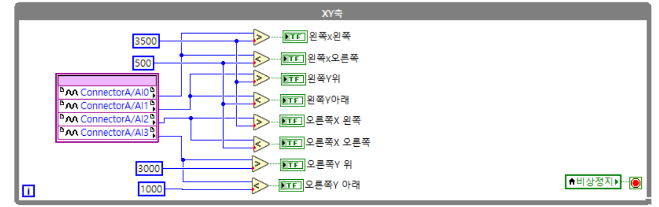
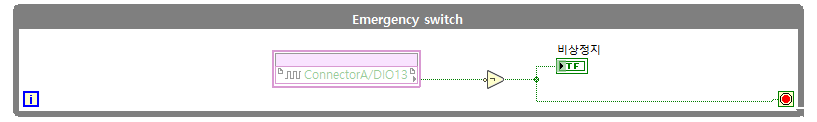
FPGA 프로그램

조종기 입력의 신호처리, 엔코더 신호신호의 4채배 분배, 모터 회전에 필요한 PWM신호 프로그램을 작성하여 생성을 FPGA에구워넣었다

첫번째, 조종기의 입력신호 처리부분





조종기는 두개의 조이스틱과 두개의 버튼으로 이루어져 있으며, 이중 두개의 조이스틱과 하나의 버튼을 사용하였다.

왼쪽 조이스틱의 위,아래의 기능은 도르래를 들어올리거나 내리기 위한 Ref값 지정에 사용된다.

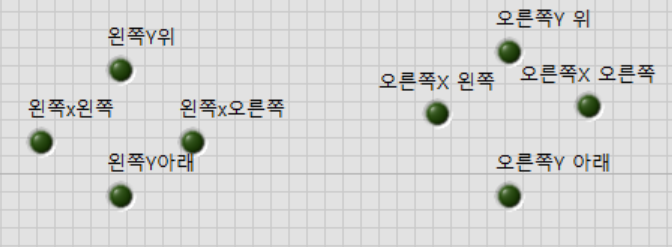
왼쪽 조이스틱의 좌우 기능은 실린더의 전진 또는 후진에 사용된다.

오른쪽 조이스틱의 위, 아래, 좌, 우 기능은 로봇의 전진, 후진, 좌회전, 우회전을 위한 Ref값 지정에 사용된다.

왼족 버튼은 비상정지 기능을 한다.

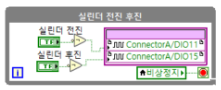
조종기는 하나의 조이스틱에서 X축과 Y축에대해 2개의 아날로그 신호를 내보내며, 0~4096의 값을 내보낸다.

버튼은 디지털 신호를 내보내며, Pull\_UP 방식으로 버튼을 누르면 0(False) 값 누르지 않으면 1(True)값을 내보낸다.



우리는 프로그램에서 조이스틱의 아날로그 값을 받아와 디지털 신호로 변환하여 조이스틱의 위치에 따라 True, False 값을 내게 하였다.

두번째, 간단한 실린더 동작부분과 모터동작을 허락해주는 Enable신호부분

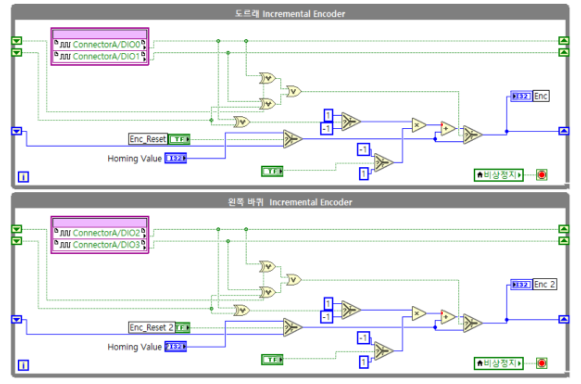


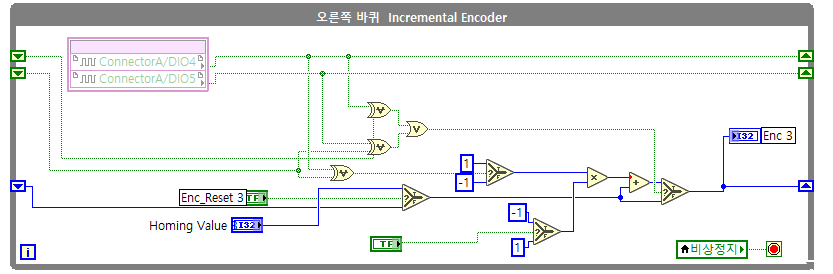
디지털 신호로 변환한 조이스틱의 신호를 받아들여 실린더의 전진 또는 후진 출력을 디지털 신호로 내보내게 하여 릴레이 모듈을 작동시켜 실린더를 움직이게 하였다.



모터를 PWM으로 동작 시킬 때 필요한 신호는 Enable 신호, PWM펄스 신호, CW/CCW방향 신호 가 필요하다. 이때 필요한 ena 신호를 프로그램상의 ena 신호가 들어오면, 모터 드라이브의 ena핀에 신호를 전달하여 준다.

세번째, 모터의 엔코더 신호 처리부분





엔코더는 두개의 신호 A, B로 두 신호 사이에 90도의 간격을 가진다.

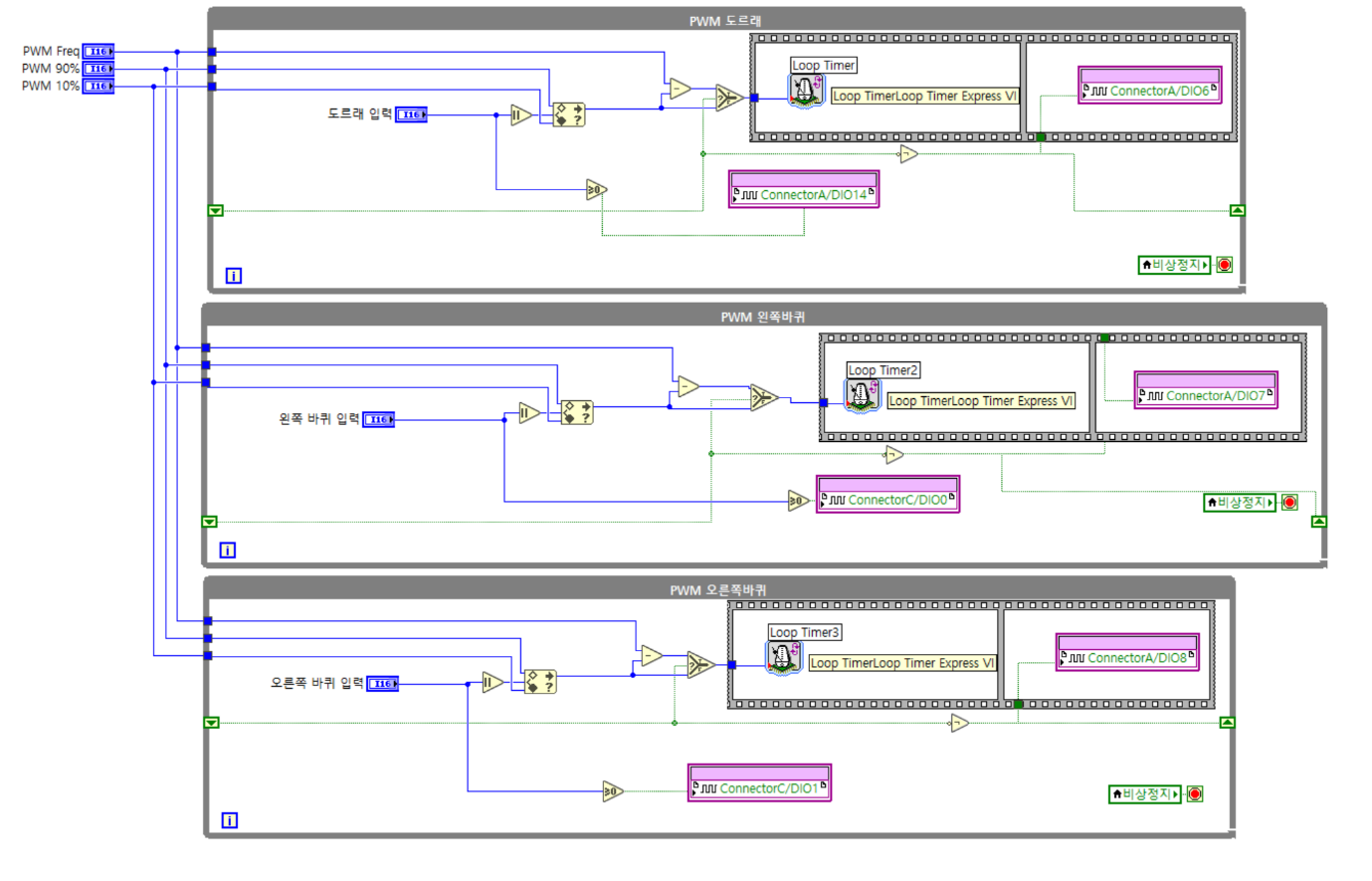
이 간격을 이용하여 모터의 회전방향과 위치를 알 수 있다.

회전 방향은 A상의 현재 값과 B상의 하나 이전 값이 다르다면 CW 방향, A상의 현재 값과 B상의 하나 이전 값이 같다면 CW방향이다. A상의 현재 값과 B상의 하나 전 값을 XOR 하였을 때 1(True)이라면 CW방향, 0(False)이라면 CCW방향이다.

A상의 값이 0에서 1로 1에서 0으로 갈 때 2번, B상의 값이 0에서 1로 1에서 0으로 갈 때 2번 270도 간격 사이동안 4번 측정하여 4채배 형식이다. 이는 각 신호 A와 A하나 전 값, B와 B하나 전 값을 각각 XOR하여 측정한다

이때 회전 방향이 CW라면 1을 곱해주고, CCW라면 -1을 곱하여 현재의 위치를 나타낸다.

네번째, 모터의 PWM신호 발생과 CW/CCW신호 부분



PWM 신호의 주기는 20,000 펄스 이다.

시프트 레지스터를 이용하여 점멸 동작을 하는 신호를 만들어 준다.

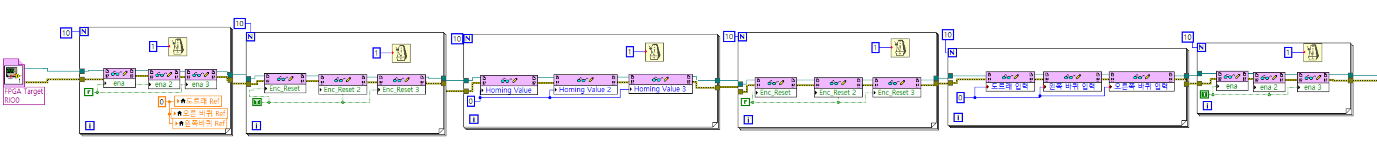
만약 모터의 입력 신호가 15,000 이라면 총 PWM펄스 20,000 중 15,000펄스동안 켜지고 5,000펄스동안 꺼져야 한다.

PWM신호를 만드는 방법은 모터 입력 값 만큼의 펄스을 ON, PWM펄스에서 모터 입력 값을 뺀 만큼의 펄스를 OFF하여 이를 반복하여 펄스를 내보낸다.

이때 만약 모터의 입력 값이 음수라면 모터의 회전방향이 반대이므로 디지털 출력신호를 이용하여 모터드라이브에 CCW방향을 입력하여 방향을 변경하여 준다.

Controller 부분

첫번째, 시스템 준비 부분



신호가 무시 당할 수도 있기 때문에 for Loop를 이용하여 10 반복하여 준다.

1번 Loop

모터의 Ena신호를 모두 OFF하고 각 모터의 Ref값을 0으로 초기화한다.

2번 Loop

FPGA프로그램의 엔코터 리셋신호을 ON하여 Homing value의 값이 엔코더 값이 되도록 한다.

3번 Loop

Homing value의 값을 0으로 만들어 엔코더 값을 초기화한다.

4번 Loop

FPGA프로그램의 엔코더 리셋 신호를 OFF하여 엔코더가 값을 받아들일 수 있도록 한다.

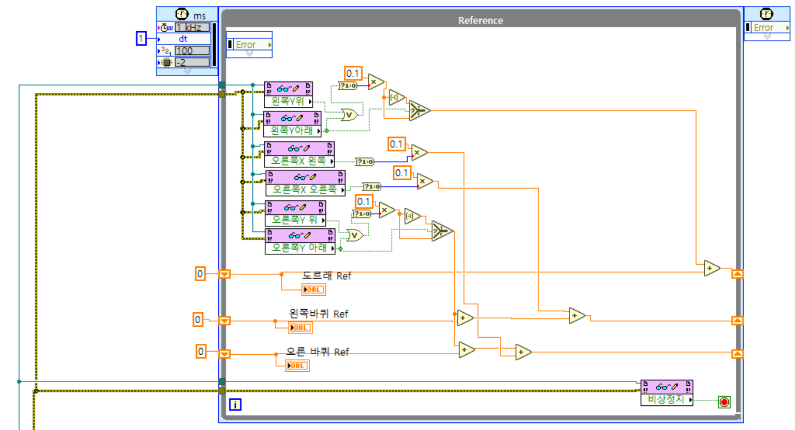
5번 Loop

각 모터의 입력 값에 0값을 주어 초기화 한다.

6번 Loop

모터의 Ena신호를 모두 ON 하여 시스템 준비를 완료한다.

두번째, 조종기의 값을 받아 Ref 값 지정



도르래 Ref값 지정

왼쪽 조이스틱의 Y측을 위로 움직였을 때 값을 Ref값 증가 아래로 움직였을 때 Ref값 감소시킨다.

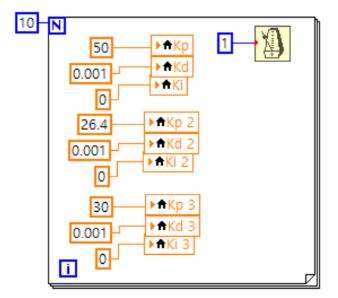
바퀴의 Ref값 지정

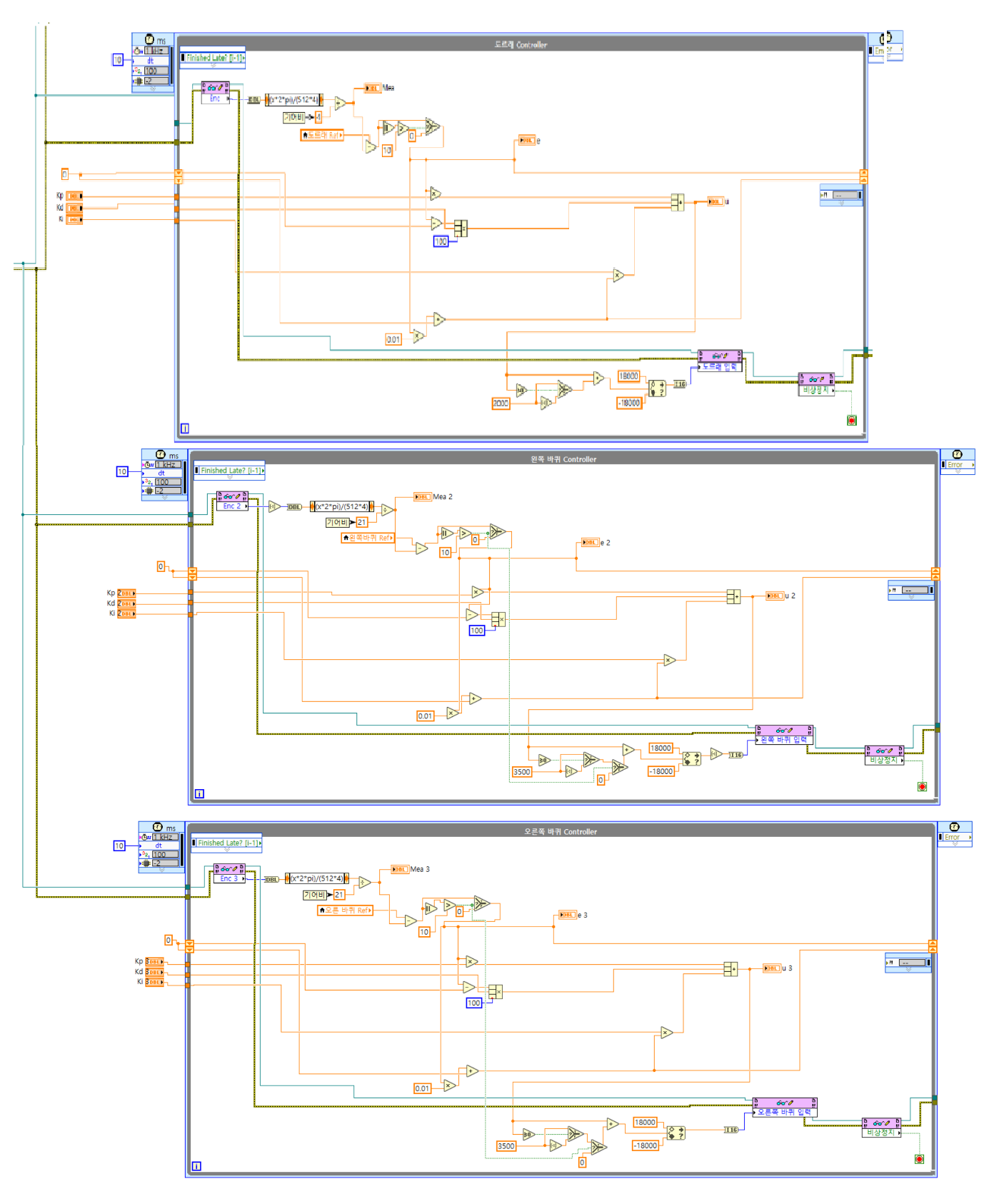
오른쪽 조이스틱의 Y축을 위로 움직였을 때 왼쪽 바퀴와 오른쪽 바퀴의 Ref값을 증가 시키고, 아래로 움직였을 때 왼쪽 바퀴와 오른쪽 바퀴의 Ref값을 감소시킨다.

오른쪽 조이스틱의 X축을 왼쪽으로 움직였을 때 로봇이 왼쪽으로 움직여야 하므로 오른쪽 바퀴의 Ref값을 증가시켜 로봇이 왼쪽으로 회전하게 한다.

오른쪽 조이스틱의 X축을 오른쪽으로 움직였을 때 로봇이 오른쪽으로 움직여야 하므로 왼쪽 바퀴의 Ref값을 증가시켜 로봇이 오른쪽으로 회전하게 한다.

세번째, PID 계수 지정과 제어 알고리즘





Error = Ref – Mea 이다.

U = P제어 + I제어 + D제어 이다.

비례(P)제어. 에러값에 비례한 출력값을 내어 에러가 클수록 큰 값을 주어 목표치에 도달하게한다.

적분(제어)… 설명

미분제어… 설명

엔코더로 측정한 현재 위치 값을 Ref값과 비교하여 발생한 Error값을 이용하여 목표 Ref값에 도달하기 위한 출력 U값을 내보내야한다.

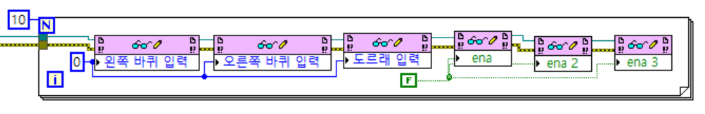
비례제어 에러 값과 P계수를 곱한 값이다.

적분제어 에러 값에 I계수를 곱하고 시간에 대하여 적분한 값이다. 에러를 누적시켜 잔류편차를 제거한다.

미분제어 현재 에러 값에 이전 에러 값을 뺀 값에 D계수를 곱하고 시간에 대해 미분한 것이다.

이렇게 구해진 값을 모두 더하여 u값을 모터의 입력에 주어 FPGA의 프로그램에 의해 PWM신호가 발생되어 모터가 구동된다. 이때 모터는 10%값 이상 90%값 이하 에서만 동작하므로 입력 값을 -18,000~-2,000, 2,000~18,000으로 제한 시켰다. 바퀴 모터의 경우 부하가 많아 최소값 2,000으로 작동이 되지않아 에러의 값에 반응이 빠르도록 최소 값을 3,500으로 변경하였다.

네번째, 시스템 종료 부분



시스템이 종료 되었을 때 마지막 값을 주지않으면 제일 마지막에 주었던 신호가 계속 보내지게 되어 예상치 못한 상황이 발생될 수 있다. 이를 방지하고자 시스템이 종료되었을 때 모터의 신호를 0으로 바꾸어 주고 Ena신호를 OFF 함으로써 시스템을 마무리한다.