```
PROGRAM Modbus2
VAR
    MB_Inputs : ARRAY [ 0 .. 99 ] OF WORD ;
    MB_Outputs : ARRAY [ 0 .. 99 ] OF WORD ;
    Huzzah_Read : BOOL ;
    Huzzah_Write : BOOL ;
    Modbus_Scan_Delay : Ton ;
    MB_Delay_Enable : BOOL ;
    MB_Delay_time : TIME := t#200ms ;
    MD_Delay_State : INT ;
    Huzzah_LED : BOOL ;
    Poll_Rep_timer : Ton ;
    MB_Poll_Rep : BOOL ;
    MB_Poll_rep_time : TIME := t#20ms ;
END_VAR
```

```
//Modbus slave Inputs
MB_Inputs [ 10 ] := IoConfig_Globals_Mapping . Huzzah_Inputs2 [ 0 ] ;
MB_Inputs [ 11 ] := IoConfig_Globals_Mapping . Huzzah_Inputs2 [ 1 ] ;
MB_Inputs [ 12 ] := IoConfig_Globals_Mapping . Huzzah_Inputs2 [ 2 ] ;
MB_Inputs [ 13 ] := IoConfig_Globals_Mapping . Huzzah_Inputs2 [ 3 ] ;
MB_Inputs [ 14 ] := IoConfig_Globals_Mapping . Huzzah_Inputs2 [ 4 ] ;
MB_Inputs [ 15 ] := IoConfig_Globals_Mapping . Huzzah_Inputs2 [ 5 ] ;
MB_Inputs [ 16 ] := IoConfig_Globals_Mapping . Huzzah_Inputs2 [ 6 ] ;

//Modbus Outputs
IoConfig_Globals_Mapping . Huzzah_Outputs2 [ 0 ] := MB_Outputs [ 10 ] ;
IoConfig_Globals_Mapping . Huzzah_Outputs2 [ 1 ] := MB_Outputs [ 11 ] ;
IoConfig_Globals_Mapping . Huzzah_Outputs2 [ 2 ] := MB_Outputs [ 12 ] ;
IoConfig_Globals_Mapping . Huzzah_Outputs2 [ 3 ] := MB_Outputs [ 13 ] ;

IoConfig_Globals_Mapping . Huzzah_Read_Trigger2 := Huzzah_Read ;
IoConfig_Globals_Mapping . Huzzah_Write_Trigger2 := Huzzah_Write ;

//send joystick value back to huzzah for use as pwm speed
MB_Outputs [ 11 ] . 0 := Huzzah_LED ;

//Read first, then write. Arduino library on supports one type at a time in a
single frame
Modbus_Scan_Delay ( IN := MB_Delay_Enable , PT := MB_Delay_time , Q => , ET => ) ;
Poll_Rep_timer ( IN := MB_Poll_Rep , PT := MB_Poll_rep_time , Q => , ET => ) ;

CASE MD_Delay_State OF

    0 :
        //Send the read request
        Huzzah_Read := 1 ;
        MB_Delay_Enable := 1 ;

        //Keep triggering the poll while scan in on
```

```
34              MB_Poll_Rep := 1 ;
35              IF  Poll_Rep_Timer . Q  THEN
36                  Huzzah_Read := 0 ;
37                  MB_Poll_Rep := 0 ;
38              END_IF
39
40              //After Scan delay
41              IF  Modbus_Scan_Delay . q  THEN
42                  Huzzah_Read := 0 ;
43                  MB_Delay_Enable := 0 ;
44                  MD_Delay_State := 10 ;
45              END_IF
46
47          10 :
48
49              //Send the write request
50              Huzzah_Write := 1 ;
51              MB_Delay_Enable := 1 ;
52
53              //Keep triggering the poll while scan in on
54              MB_Poll_Rep := 1 ;
55              IF  Poll_Rep_Timer . Q  THEN
56                  Huzzah_Write := 0 ;
57                  MB_Poll_Rep := 0 ;
58              END_IF
59
60              //After delay
61              IF  Modbus_Scan_Delay . q  THEN
62                  Huzzah_Write := 0 ;
63                  MB_Delay_Enable := 0 ;
64                  MD_Delay_State := 0 ;
65              END_IF
66
67      END_CASE
68
```