RENEWING A CAR TRACK, CONTROL PANEL AND PROGRAMMING WITH SIEMENS

HAMK
UNIVERSITY OF APPLIED SCIENCES

Bachelor's thesis

Automation Engineering

Valkeakoski, 5 May 3013

Jeevan Adhikari

Author Jeevan Adhikari **Year** 2013

Subject of thesis Renew the Car track, control panel and programming with Siemens

ABSTRACT

This research examines use of automation controllers for the solution of automation problem. The automated car track at the HAMK automation laboratory was used as an example for implementation. The commissioner of the thesis was the Department of Automation, HAMK University of Applied Sciences. It is located in Häme region in southern Finland having 7 campuses in different locations.

The objective of the study was to build a PLC program that could control the speed of the car. The other objective was to build a HMI so that the process happening at PLC could be monitored from control panel. The control panel is used for reading or writing variables in PLC. This way we could control the whole system from the control panel. In order to implement the task, the knowledge of STEP 7 program and technology objects and function within the S7 CPU was required. Interface programming skills was also essential in order to design the control panel. Understanding of working principle of amplifiers, transformers, proximity sensors, resistors was also needed.

First approach was made to understand the commissioner's problem. And then the information (manual, data sheet, and working principle for example) about the components being used was gathered. The technical information was gathered accessing online webpages of manufacturers. Written description of the solution, the SA model was for making contest diagram and first level data flow diagram. After written description of the system, STEP 7 programming was done. The HMI programming was done using WinCC flexible and then was integrated with the STEP 7 project.

The car track was controlled automatically. The process could be controlled from the control panel and monitor the variables form it. The approach to measure the speed of car at every segment of the car track was done. The speed measurement function was partially successful. Other functions like time measurement and counting the number of laps completed by both cars was also done. The benefits to the commissioning organization were that it received a working car track control system. The main benefits to the researcher were researcher understood overall automation concept design and implementations, design of human machine interfaces.

Keywords Automated car track, PLC programming, HMI, STEP 7, WinCC flexible, Automation design

Pages 28 p. + appendices 36

CONTENTS

Some abbreviations

CPU	Central processing unit
‘A sensor’	Sensor at starting position, automated car side
‘M sensor’	Sensor at starting position, manual car side
HMI	Human machine interface
IO	Input/output
PLC	Programmable logic controllers
Wait for start	The initial state of car track
Count mode	The counting situation before the race begins
Run mode	The state when cars are racing

1 INTRODUCTION

The objective was to program, build control panel for an automated slot car track. The two tracks, one controlled by human manually, and the other one controlled automatically by using programmable logic controllers. It requires that position of cars to be detected with contactless sensors. Using a PLC the speed of car is controlled in correspondence to position of car. An HMI (human machine interface) enables to monitor the state of car track and start up and stop decisions by user. The PLC continuously measures the sped of car and shows the value in control panel when the car automatically working is running. HMI also shows number of laps car has completed in real-time.

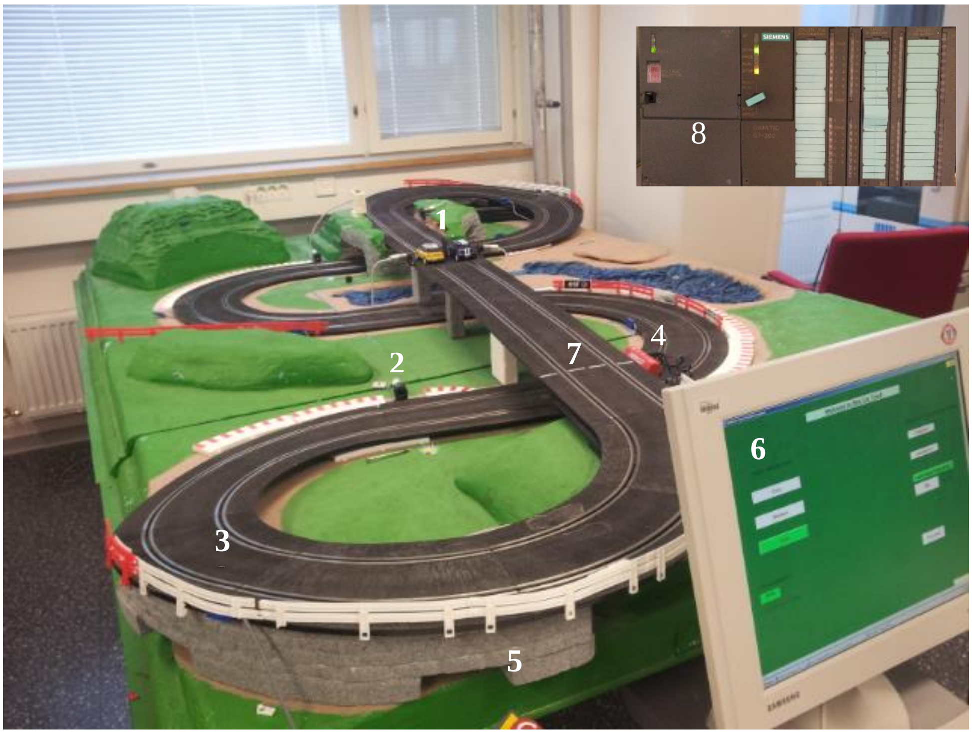


Figure 1 Slot car track layout

1. Cars at starting positions
2. Photoelectric proximity sensor
3. Magnetic proximity sensor
4. Power to the manually controlled car
5. Emergency push button
6. Control panel
7. Racing track
8. PLC station

2 UNDERSTANDING THE AUTOMATION CONCEPT

The general idea of using an automation concept is to supply required DC voltage to the track in corresponding to the car's position. The car track is controlled with Step 7 automation technology. The station in use is s7-300 family. This family is suitable for number of IOs we have in our control system. The task of building whole automation system includes hardware installation, PLC programming and programming for HMI. A Human Machine Interface (HMI) device is used for controlling and visualization of the process.

2.1 S7 300 Station

To understand how this control system works, it is important to know how programmable logic controllers (PLC) operates. A PLC 300 station has following components in use.

Table 1 List of automation modules used

Device	Model	Additional information
Power supply		
CPU	312 IFM	DI10/DO6×24VDC
Digital input module	SM 321	DI 32×24VDC
Analog output module	SM 332	4×12Bit

A programming device, A PC in this case has been used for programming S7 300. A programming device does initialization, configuration and test functions. The S7 300 station has been connected with programming device with a special cable called MPI cable. While networking different components in automation or IT systems, we need to define a node address for each devices. Here the address for programming device was defined as 0 and address for S7 station was 2. Following figure show the connection of various S7 station components and programming device.

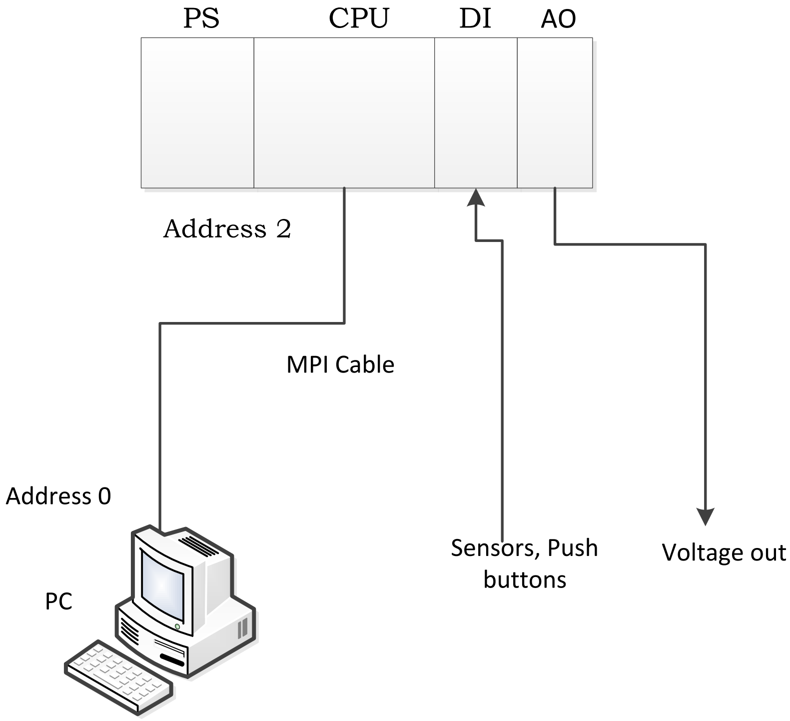


Figure 2 Schematic diagram of S7 300 station

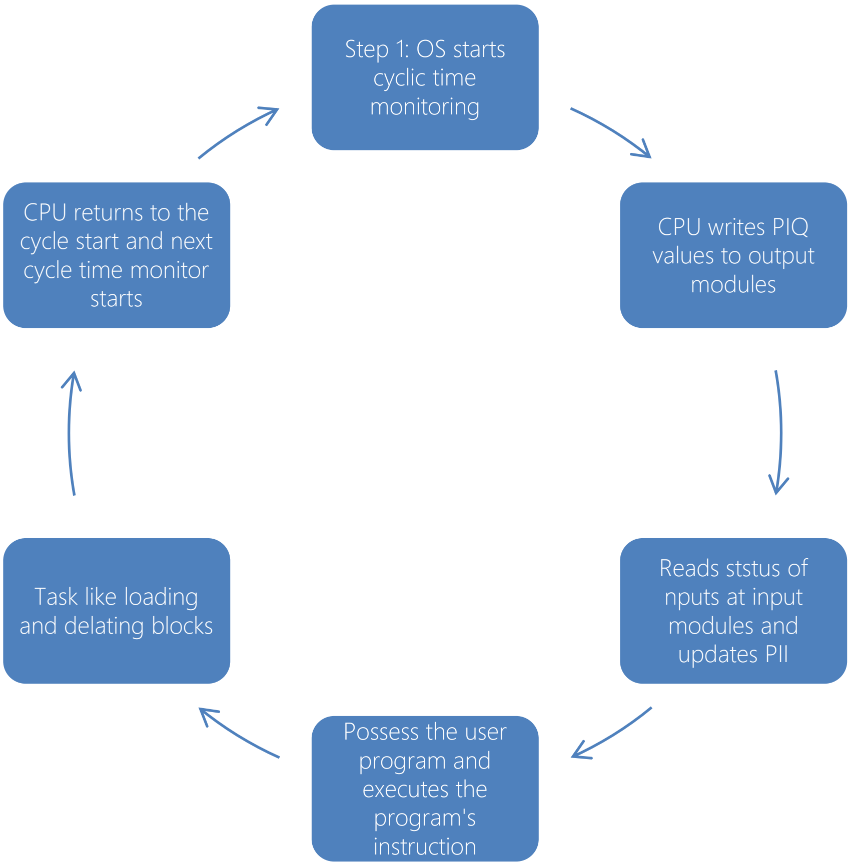
2.1.1 CPU

The function of the CPI is to execute user program and to communicate with all the nodes connected with network.
CPU 312 facilitates MPI interface functionality. Most of the CPUs build nowadays have Profibus DP master or/and Profinet master system. Before a CPU can communicate with a programming device, PG/PC interface setup was done. The transmission rate up to 12Mbps can be selected but max of 187.5 Kbps was possible with CPU used in the station.

This CPU does not have hardware clock (integrated 'real time clock'). The software clock was used instead. This specific CPU contains special integrated function; process interrupt, counter and frequency. As the CPU does not have any real time clock so it lacks accumulator battery. Processing time for bit instructions is 0.6 µS, word instructions 2µS and double integer math takes 3µS. The nesting depth is 8, it can handle only 32 FC/FB. It also supports IEE timers and counters.

Cyclic program processing by CPU

Following figure shows the sequence how a CPU operates.



Cycle time of CPU

The cycle time represents the time the operating system requires for processing one program cycle, that is one OB1 cycle

For calculating the process image (PI) transfer time;

Transfer time = Base Load (K) + Number of bytes in PI in rack 0* A

In current station,

K= 140 µS for CPU 312 IFM
A= multiplication factor per byte

So, transfer time for PI= 150µs+ 5×37= 335µS

By monitoring CPU program execution time form STEP 7, the program execution time is 4ms
Multiplying the value by factor 1.06 which is for CPU 312

The program execution time = 1.06×4ms
=4.24ms

The transfer of process image output = 150 µs+ 1×37= 187 µs.

Cycle time (neglecting time for S7 timers and communication vie MPI),

Cycle time = 4.76ms

2.1.2 Signal modules

Digital input

The SIMATIC digital input module has thirty two isolated DI in group of sixteen. Rated input voltage is 24 VDC. This type if signal module is suitable for two, three or four wires proximity switches.

Analog output

The function of analog output module is to convert digital output values received from CPU to analog signals. In the system used in car track, analog module gives output current signal range 0-20mA. An appropriate calculation was done in order to supply required supply voltage to the track. This is done by moving decimal values form range to the analog output address.

Table 2 Analog output range in S7 module

	Current output range	
Decimal value	**0mA-20mA**	**Comments**
32726	23.96	Over flow , out of range
32512		
27648	20mA	Rated range
20736	15mA	
1	723.4nA	
0	0mA	
-1	-3.52mA	Under the range, not in USE
-4864		

The module has four output ports, but three of them are deactivated from programming interface. Only one is use for supplying current. This specific module has options to choose as current or voltage output. The current can be chosen either 4-20mA or 0-20mA. 0-20 mA output range was chosen.

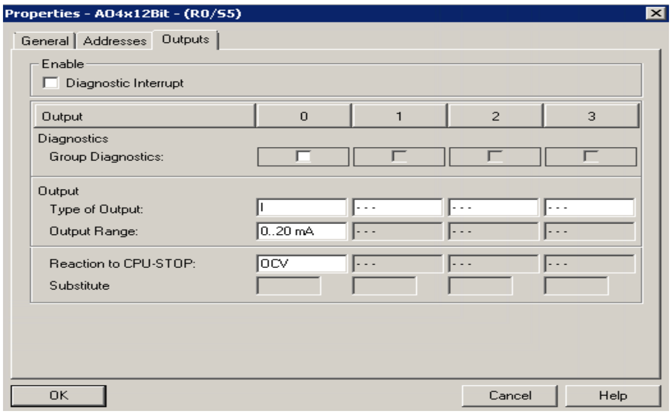


Figure 3 Analog output module properties setting

The I/O is written on demand of program, via peripheral with PQW. IN FBD programming language MOVE command was used to write analog value. (Transfer operation should be used whiling using Statement List (STL)

2.1.3 Power supply

Integrated IO module with the CPU, digital module (SM 321), and analog output module (SM 332) gets power from power supply PS 307. There are 3 sets of terminals for 24VDC output. The mains supply of the PS 307-5 A power supply is protected with a miniature circuit-breaker. Protect the mains with a MCB rated current at 230 VAC 6A and tripping characteristics type C

2.2 Sensors

The cars running on the track are to be detected when they move through sensors. If an object comes into the detection range of a sensor, this causes a signal change from 0 to 1 at the switching output of the sensor. For tracking the position of cars sensors are placed in different location in a track. All together six photoelectric and 14 magnetic proximity sensors are used. A photoelectric sensor detects the car when it is entering a curve. Magnetic proximity sensor detects weather the car is at the beginning, or in the middle or at the end of curve. All the sensors are discrete inputs and sourcing.

2.2.1 Magnetic proximity sensors

This sensor is able to detect objects with magnetic property without any touch. For this reason magnets were kept inside the cars. These sensors were used because they have long functional life and are comparatively cheaper. All the sensors are connected as shown in following diagram.

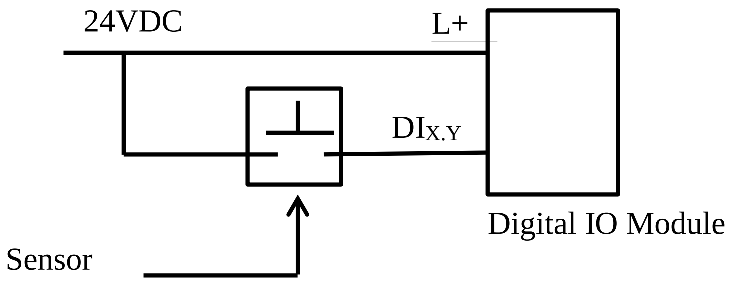


Figure 4 Sensor connection with s7 input module

L+ supply 24VDC to sensors and in $DI_{X.Y}$, X refers to byte address and Y refers to bit address of a signal.

2.2.2 Photoelectric sensors

These devices are used for detecting the presence or absence of an object by using a light transmitter infrared and photoelectric receiver. This sensor has four wires. Two of the four wire sensor is connected to a power supply from which sensor gets its power, and the two other are used as separate outputs, each of which is connected to load. Following table shows how sensors are connected with input module.

Table 3 Connection of four wire sensor according to European standard

Brown wire	+VDC supply (L+)
Black wire	Load(input channel normally close
White wire	Load(Input channel normally open
Blue wire	Neutral (M)

2.3 HMI

HMI is programmed with a Simatic WinCC flexible. The process is visualized by Simatic WinCC runtime system on a PC. Fundamental requirements for a HMI are,

- Start and stop commend for race
- Select number of laps
- Select different difficulty levels
- Visualization of automated and manual car's performance (number of completed laps, time span, record of best race
- Speed of automatically controlled car

2.4 Amplifier

The analog output can only supply 0-20mA current range. This is not enough for supplying enough voltage to the track. To get appropriate amount of voltage an amplifier is used. This is done by adding a resistor and transistor in the circuit.

When value of resistance is added to circuit, current remaining constant, the voltage is the circuit increases. This can be verified using ohm's law as shown below,

$$V = IR \tag{1}$$

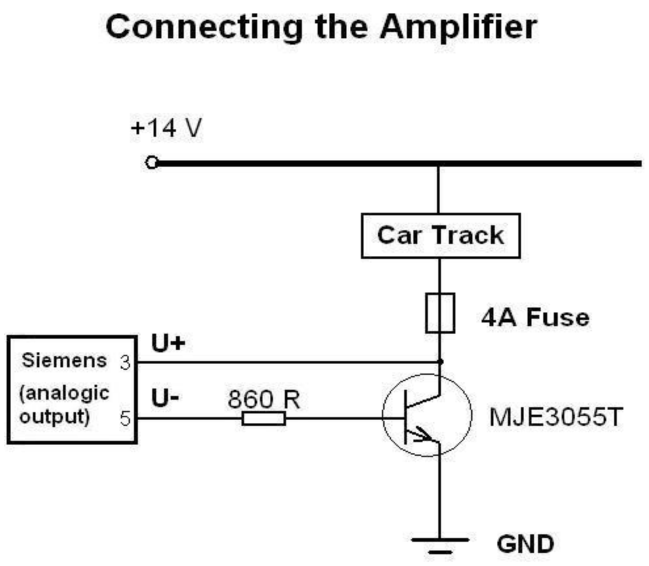


Figure 5 Connection of amplifier to AO module

2.5 Required software components

- STEP 7

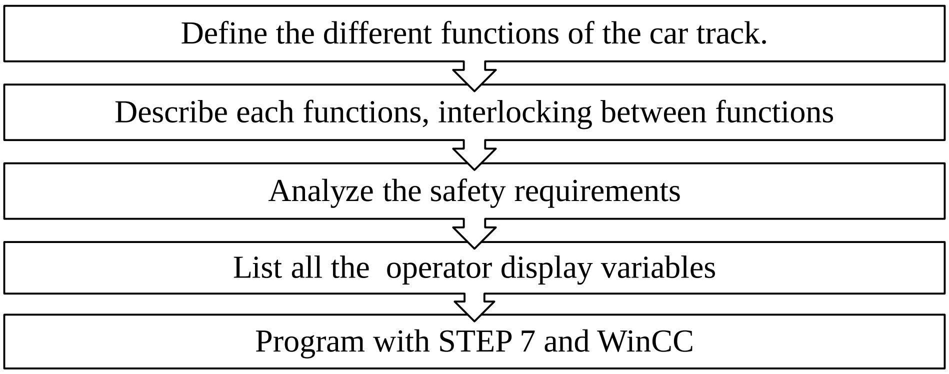
It is standard software package from SIEMENS for configuring and programming SIMATIC PLC. It includes components like SIMATIC manager, Symbol editor, Hardware configuration, Hardware diagnostic, NETPRO communication configuration, Programming languages. The function of SIMATIC manager is to manage all the data that belong to an automation project. The symbols are managed with symbol editor. Three programming languages ladder logic, statement list and function block diagram are integral package of S7 300. In the car control system function block diagram has been used as coding language. Hardware configuration is the tool to assign parameters to the hardware. The examples of hardware could be central or distributed I/O, CPU, and CM etc. The communication is configured using NetPro in STEP 7.

- WinCC Flexible 2008

This HMI programming tool includes three components, WinCC flexible engineering system, and WinCC flexible runtime and WinCC flexible options. The WinCC flexible engineering system handles all the essential configuring tasks. And the Runtime system is for process visualization. It executes the process in the runtime. WinCC flexible runtime has enabled to use PC as an operator panel.

3 EXAMINING THE AUTOMATION CONCEPT

Before programming a PLC program, a plan was made for automation concept. This was made for simplicity of overall coding of program.



3.1 Defining the different functions of the car track

The overall function of the control system is to supply appropriate voltage to the car track. The amount of voltage supplied is depending on the difficulty level chosen by the user. So program has been designed in such a way that it can handle three difficulty level. The other function is to measure the speed of the car. Below the list of all the functions car track does are mentioned.

- Adjust the speed
- Detect car's position and supply appropriate voltage
- Measure time durations
- Measure time span for current laps and all laps for both manually and automated car.
- Count the number of laps
- Parameters set in wait for start
- Parameter set in the count mode
- Run mode
- Measure the speed

3.2 Functions, interlocking between functions

When the CPU is changed from stop to run mode, the track is at the stop mode that means it is waiting some commands in order to start the process. When start button is pressed from control panel, the countdown should begin. During the stop mode and countdown period there should not be voltage supplied to the track. The end of countdown enables the track in the run mode. In this mode only automated and manually controlled car can start the race.

The voltage in the manual track is controlled by user by changing a resistance of a variable resistor. In case of automated track voltage is supplied by control system depending in position of car. The speed of car on the highway is made comparatively greater than turning positions. When the car is entering the curves speed is decreased, during the curve it remains constant, but at the end a curve speed increases slightly until it reaches another curve. The same principle applies to three other curves.

Three states are used for controlling the car track. The initial state is the stop mode. This is the situation when controller is changed from stop to run mode. At this mode the red led indicator is on, indicating no race allowed this moment. Zero voltage is supplied to the track.

The speed of the car differs on every segment. The speed of a car at a certain segment has to be updated immediately after car leaves that segment. The value of speed has to be updated when there is change in the speed.

3.3 Descriptions the safety requirements

The possible risks were analyzed and necessity of additional protective measures was detected. The additional protective measure shall protect the user from any hazards that may exit. At first risk assessment process was done, If the risk assessment process is not adequate, then risk reduction process is done. But in this system risk assessment process was enough to deduct possible risk. But there is always residual risk in any systems.

Risk assessment

Risk assessment was started with defining the functions of whole system.

After the defining of the function of the system, the most important step, identification of hazard was done. The possible hazard in this system could be mechanical damage on the conductor on the track, overheating of transistor, short circuit. There was also hazard due to material and sub-stances.

Risk elimination and risk evaluation

Risk has been evaluated using equation

$$Risk = Consequences \times Probability\ of\ occurance \qquad (2)$$

The voltage supplied in both manual and automated has to be zero in case of emergency situations. This stops the car immediately. To eliminate the risk from electrical short circuit normally close pushbutton has been mounted on the side of operator’s display. Upon pressing this button, the HMI was programmed to display the notification of emergency situations. Red colored window which actuates in hazardous situation or emergency.

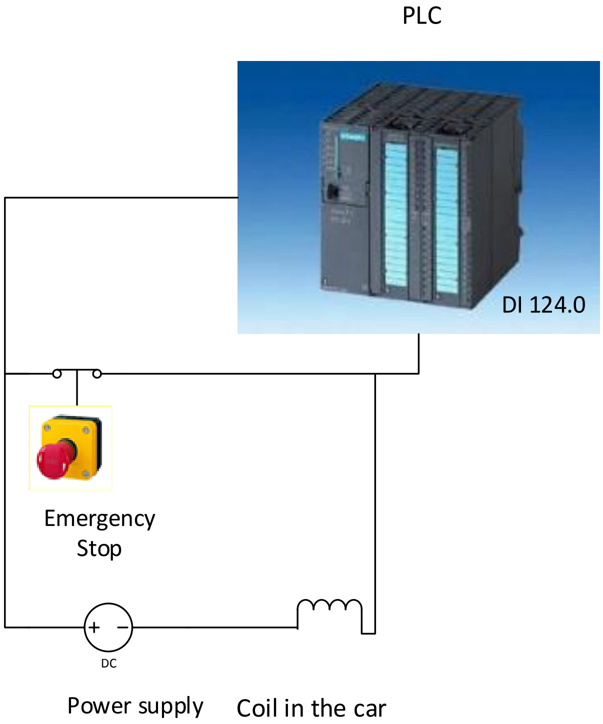


Figure 6 Mounting of emergency push button.

3.4 List of all the operator display

Screen requirements

a. Start screen

- Select NO. of laps or select continuous run options or time span
- Select difficulty level
- One of the three bit should be high during the run mode, enable high when click, reset other bits.
- Instruction to place both cars on start position
- Monitor the state of the car track
- Monitor the state of CPU
- Alarm window in case of emergency situations
- Proceed to another screed when suitable parameter setting is done

b. Race screen

- It shows the visualization of the process showing the real time
- State of the car track
- (Red color for stop mode, yellow for count and green for run)
- Assign colors with state
- Start race button
- NO. of laps completed by both manual and automated
- Overall time for certain race

- Time taken for specific laps
- Speed of automated car in real time
- No. of remaining laps for both cars
- Emergency alarm when E. Stop button pressed
- Stop button
- Exit visual runtime system
- When Manual car start before run mode is allowed, then error message is displayed in the HMI device. This situation doesn't allow setting the run mode. While run mode is going to be activated M sensor should be high during that case. After placing the car on starting position counting mode is again activated.
- Alarm window in case of emergency situations

c. Result screen

- When automated or manual car first finishes a race, result of race appears in new screen. This screen shows;
- 'You win' notification when manual car first finishes the race. 'You lose' when automated car finished the race first.
- Shows overall time for both race
- Shows laps time for all round both manual and automated car's.
- Shows top record for manual car
- Option to proceed to other screen or exit the runtime system.
- Alarm window in case of emergency situations

d. Monitoring screen

- Monitor the states of all the sensors, when the sensor detect an object in its range, it should be visible in the HMI.
- Option to go back to previous screen or exit the runtime system.

4 PLC PROGRAMMING USING STEP 7

4.1 PLC programming overview

The S7 CPUs comprises the operating system and the user program. The operating system does functions like updating the PIE and PIO, call the user program, management of memory areas, handling restarts etc. The operating system parameters can be changed in some situations, but the parameters were leave default in the car track control station. The other part of the program is user program. This contains all the logical functions and calculations required for processing user program.

The user programs were created and loaded to the CPU. The programming method so called structured method was used for breaking down the program into individual, self-contained program sections. Such structured programs contains different blocks for various purposes. Deciding how each blocs are called for processing is an important step in STEP 7 programming. The conditions and how often the block is called was defined beforehand.

Normal cyclical processing of STEP 7 program is handled by OB1. After power is applied and the CPU is switched to RUN, OB1 is called and processed on each CPU cycle, until the CPU is stopped or power is removed. Other blocks like FBs, FCs, SFCs and SFBs are called directly or indirectly from OB1. During one cycle time OB1 is processed form first network to last. If the calls are made from OB1 to other blocks, as each block completes its execution, control is back to OB1. In the user program of car track only start-up block and main program were used. Other organisation blocks like Time-of-day interrupts, Time-delay interrupts, Cyclic interrupts, Hardware interrupts, Multi-computing interrupts, Redundancy error interrupts, Asynchronous error interrupts, Synchronous error interrupts were not called by operating system.

Global symbols were used for addressing inputs, outputs, timers, counters and some other memory addresses. In most of the blocks, block-local symbols were used. These types of symbol are valid for only one block and are declared in the local data. Absolute addressing was also used for positive or negative RLO edge detection.

4.2 Hardware setup

In Hardware setup process was done to assign the addressed used in the program to the modules. All the IO module were centrally located, no any distributed IOs used. Most of the CPU parameters were kept default.

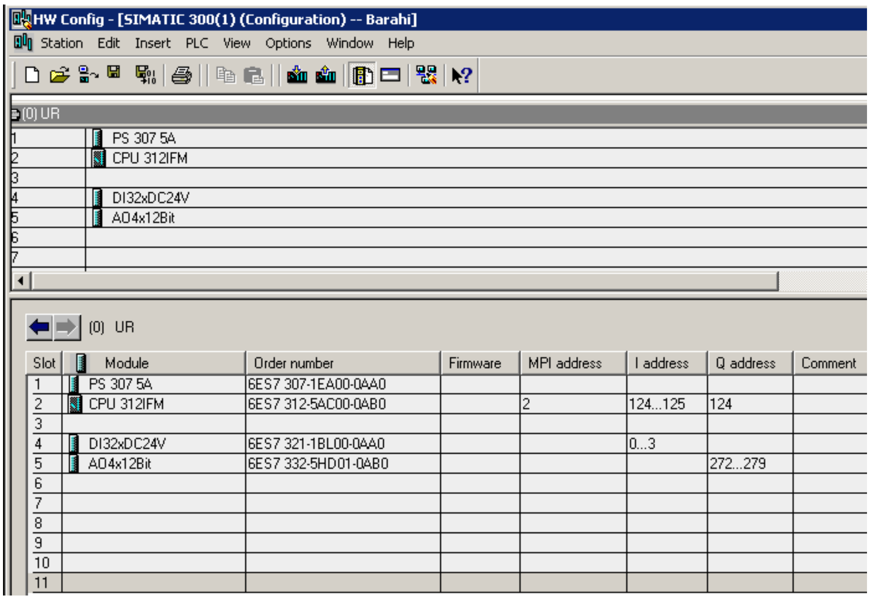


Figure 7 Hardware configuration window

4.3 Communication setup

PC interface with the PLC. This has been done in order to setup the communication between PLC and the programming device. The monitoring of the modules, downloading/uploading of the user program is only possible after setting up this interface. Interface type was MPI auto adapter, which has capability to communicate with speed 187.5kbps.

4.4 Program structure

Sequential function charts (SFC) had been created used for programming simplicity. A flow chart is a chart that flows from one stage to next and it shows what stages or events is first, second etc. under what conditions.

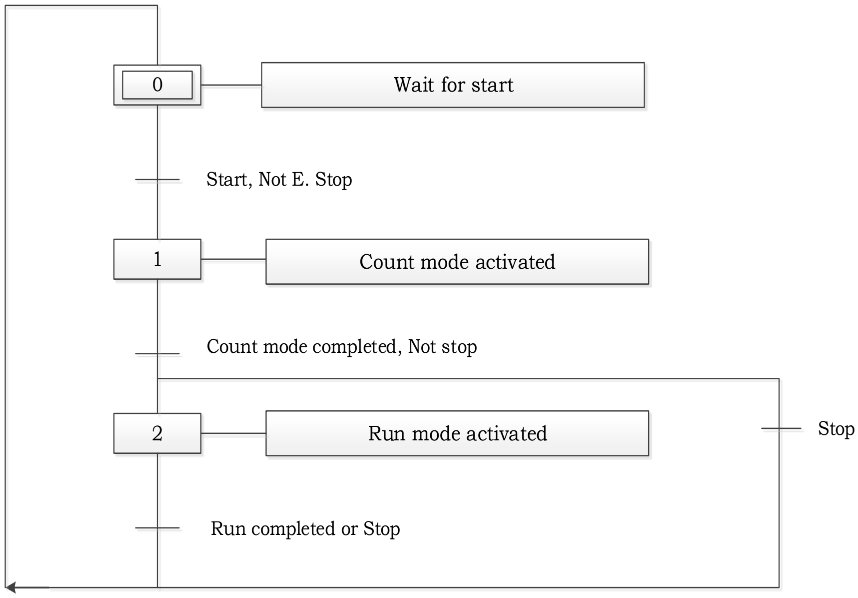


Figure 8 SFC of automation concept

4.5 Use of data types

The data were organized and defined. In STEP 7 each variable requires a data type declared. A variable's data type is depending on its width and how the binary data is represented and interpreted. We can divide the data types into elementary, complex and parameter data type.

Elementary data types

Elementary data type are BOOL, BYTE, WORD, DOUBLE WORD (DWORD), INT, DINT, REAL, DATE, TIME,CHAR, S5TIME, TIME-OF-THE DAY. The data type BOOL was used for example to represent switches in the HMI, which either can be true or false. (The address to represent BOOL address could be M0.0, 124.0). The data type word contains 16bits. A word data can be used in several forms. In the car control system, WORD was used for containing various data types, for example WORD format was used as unsigned decimal to represent number of laps completed by car. It can also be specified as a BCD and 16-bit string.

Data type DWORD has 32 bits. An INT data type has 16 bits, as is signed decimal value. The INT data type was used to input the value of number of lap user wants to play. The value can be written on the screen in the HMI. The range for INT data type is -32768 to +32767. The fifteenth bit represents the sign of actual value. 0 in that bit place means the positive value and 1 means negative value. Data type DINT was also used frequently in programing of car track system. To represent the time value in seconds or in millisecond DINT format was used. This type of data type reserves 32 bits of memory. It represents signed decimal value as INT but the range is higher. The representation of floating point numbers in quite common in mathematical calculations as well as measurements of process values. For this type of format REAL data types are used. Whiling calculating the speed of the car, real data type format was used. The reason was to achieve better accuracy. REAL variables have 32 bit floating point format. The bit from 0 to 22 determines mantissa and remaining bits 23 to 30 determines exponent. The zero value in bit 31th place means the value is positive and 1 means negative value.

Another data type is DATE. It requires 16 bits memory, The IEE data format is D#XXXX-XX-XX. No such data type has been in use in car control system. To represent the time value with a resolution of 1ms, the IEE data type TIME is used. It reserves memory of double word. The format is TIME#XXDXXHXXMXXSXXMS. The TIME format data type has been used for saving the time stamp of CPU is various situations. By comparing two time stamp, the duration of certain process has been calculated. For example, duration of race in the car track system.

S5TIME was used as preset value for timer. The value for time has been given in the format S5T#XXHXXMXXS. The bit from 0 to 11 determines the time value, 13 and 14 bits value determines the time base. Another data type CHAR reserves eight bits of memory. It is used for representing the ASCII character.

Complex data types

Complex data types were used for defining variables of different data types. A data block has been created by combining different data type like TIME, STRING. Different data types were also specified as temporary variables in L-stack.

4.6 Structure of the S7 Program

The call hierarchy of the blocks.

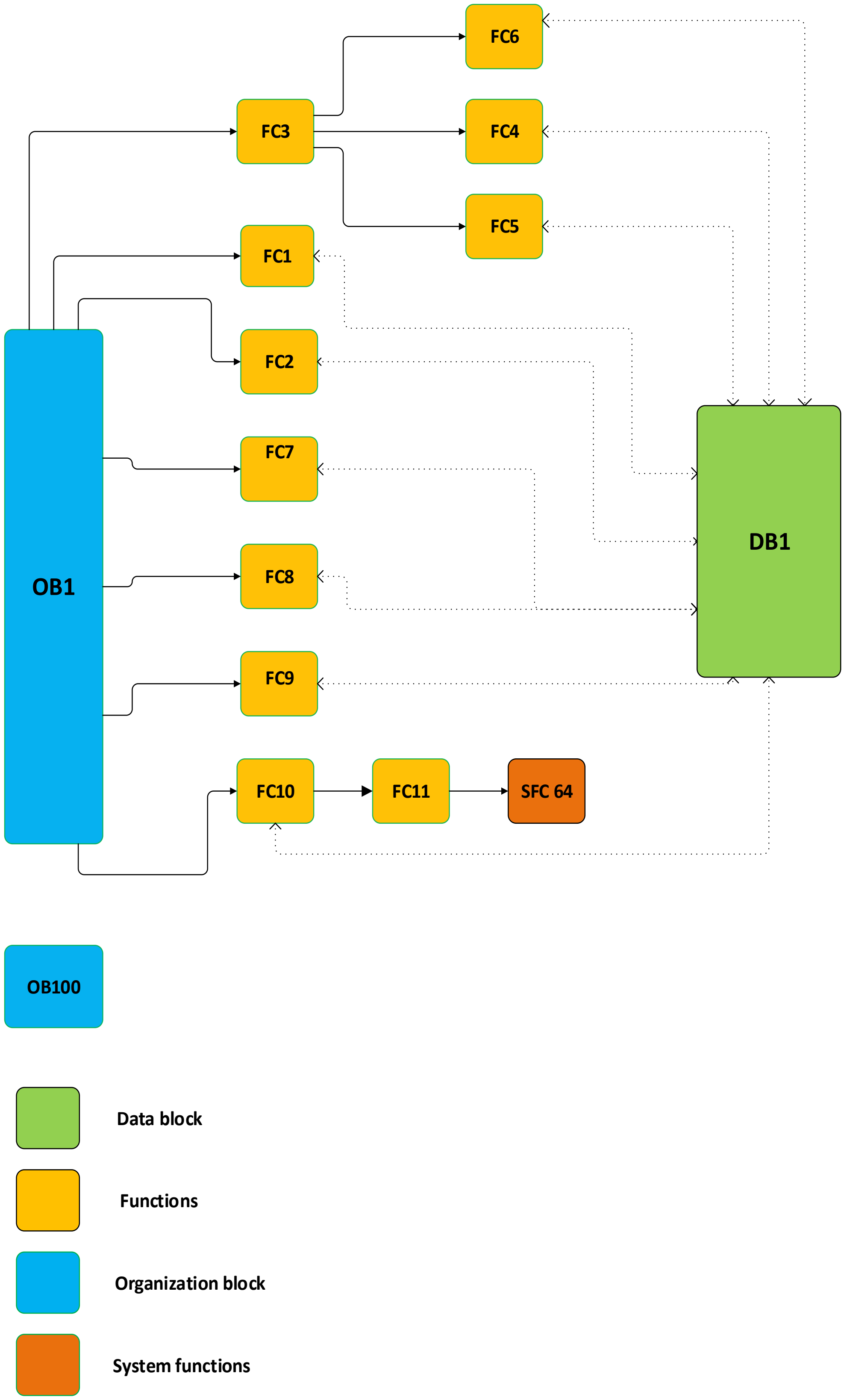


Figure 9 Order and the nesting of the block calls (call hierarchy)

4.7 Brief description of the individual software blocks

- OB1

  This OB is used for cyclic program processing. All Other blocks which were created by user were called directly or indirectly at OB1. The main sequence of the control task has been programmed at OB1.

- OB100

  This program in OB100 is executed only when restarting the CPU. Re-starting can be done by switching power on or by switching stop mode to run mode in the CPU. The OB100 detects restart of PLC and sets a bit in internal data like data block or in the memory addresses. In the car track, a specific memory which represents the wait for start is set at startup. Other memory which represent the car track in count mode and run mode has made reset The OB100 hasn't been called anywhere because it is part of operating system of CPU.

- FC1

  This user function defines the situation when the track in at stop mode. When waiting mode is active, a red signal is ON at control panel and also the LED display assembled locally. The potential difference in the track is lowered to zero so that car doesn't move at this state. This was done by moving 0 integer value into PQW address of analog output.

- FC2

  This user function counts before racing is started. Three yellow LED lights are on simultaneously. For counting down three On-Delay S7 timers have been used. When the counting is finished, all yellow LED are reset. The pressing of emergency button at this state sets car track to the wait for start state.

- FC3

  The user function realizes the situation when the car is at run mode. Green led and signal is blowing in operator's display. The blocks controlling the speed of three different difficulty level selections are called by FC3. The block is called only when corresponding difficulty level is selected. That means the call was conditional.

- FC4

  It determines the voltage output in correspondence to the car's position when easy difficulty level is chosen from the control panel.

Table 4 Integer value moved to PWQ at easiest level

Position	Integer value	Voltage in the track
Starting position	13500	
Curve 1 entering	12000	
Curve 1 middle	12500	
Curve 1 end	13500	
Curve 2 entering	12500	
Curve 2 middle	11500	
Curve 2 end	13500	
Curve 3 entering	12500	
Curve 3 middle	11500	
Curve 3 end	13500	
Curve 4 middle	13000	
Curve 4 end	13500	

- FC5

  Medium difficulty level

  It determines the voltage output in correspondence to the car's position when medium difficulty level is chosen from the control panel.

Table 5 Integer value moved to PWQ at medium difficulty level

Position	**Integer value**	**Voltage in the track**
Starting position	14000	
Curve 1 entering	13000	
Curve 1 middle	13500	
Curve 1 end	14000	
Curve 2 entering	13000	
Curve 2 middle	13500	
Curve 2 end	14000	
Curve 3 entering	13000	
Curve 3 middle	13500	
Curve 3 end	14000	
Curve 4 entering	13000	
Curve 4 middle	14000	
Curve 4 end	14000	

- FC6

Hard difficulty level

It determines the voltage output in correspondence to the car's position when hard difficulty level is chosen from the control panel.

Table 6 Integer value moved to PWQ at tuff difficulty level

Position	Integer value	Voltage in the track
Starting position	16000	
Curve 1 entering	15000	
Curve 1 middle	14000	
Curve 1 end	16000	
Curve 2 entering	15000	
Curve 2 middle	14000	
Curve 2 end	16000	
Curve 3 entering	15000	
Curve 3 middle	14000	
Curve 3 end	16000	
Curve 4 entering	15000	
Curve 4 middle	14000	
Curve 4 end	16000	

– FC7

It determines the number of cycles manual and automated car has completed after the race has begun. Automated car counter is counting up from zero at positive edge detection of 'A sensor' When run mode is initiated counter value is preset to zero. Current counting value is saved to data block and shown at HMI control panel in integer format.

– FC9

This function block has been used for calculating loop time. When the cars initiates to race from initial position and come back to same position, then it completes a loop. FC9 continuously measures the loop time and saves the time value in millisecond by using double integer data type. For measuring the time duration in S7, SFC 64 has been called two times. By calling SFC 64, it records the system time of CPU in TIME format. To measure the duration of loop, comparison is done on the result of SFC 64 calls. In HMI the millisecond value has been displayed in seconds by using linear scaling functions.

For measuring a loop time for automated car, time stamp of CPU is saved to temporary memory stack while there is negative edge detection at 'A sensor' or run mode is initiated. Second call of SFC 64 saves continually

CPU time until run mode is active. The difference in time value at first and second call defines the loop time.

A similar function is used for calculating overall time of race. Overall time means duration when run mode is activated until user wants to finish the race.

Other way of measuring time duration is S7 controller is use of clock memory. Clock memory is bits which periodically change their binary values. A period duration/ frequency are assigned to with each bits of the clock memory byte. For example 5th bit place of byte produces a frequency of 1Hz (Time duration 1sec). An up counter can be used for measuring time, by counting positive edge at clock memory bits.

By using ON-delay or OFF-delay Simatic timers, the time duration between two events could be measured.

Figure below is the simulation window's screenshot showing the time duration measured using different methods. A separate S7 project was created to illustrate this.

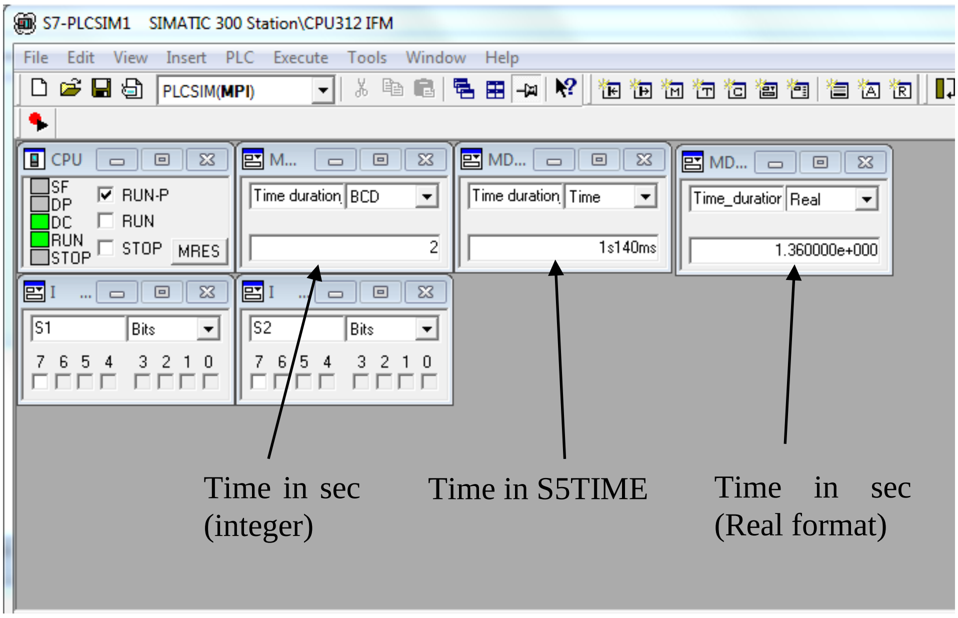


Figure 10 Illustration of three different time measurement

– FC10

This function has been used for calling ''speec_calc'' block. The same block has been called several times in order perform one function for several times. On each call the different addresses were given at input interfaces and the block's output interface gives speed of car at various segment. This ways of programming makes programming shorter.

– FC 11

In order to calculate the speed, the time taken to travel specific distance has to be measured. In this situation the distance between the sensors in fixed. The distance between the sensors was measured appropriately and measured values were supplied to S7 in standard SI unit. To measure the time, the time stamp of CPU was saved to memory place when the first senor's status goes high. Again when the second sensor's status goes high, time stamp is saved to another memory place. The difference between two time values was calculated using DIF_DI. This gives the time value in milliseconds. The time was converted to seconds.

$$Speed(S) = \frac{Distance\ between\ senors(m)}{Time\ taken(s)}$$

For handling the temporary local variables declared in the block, local memory (L-stack) in the CPU was used. The temp variables assigned in variable declaration table are used only at a time when a block is being called.

5 DESCRIPTION OF USER INTERFACE

HMI software was used for configuring the user interface for the controlling systems. The programming of HMI was done using Simatic WinCC flexible, and the process visualized with Simatic WinCC runtime system. The WinCC runtime system communicate with the automation systems, does on screen visualization of the images. It gives values for the memory uses in the PLC. This software contains most of the configuration data for HMI device. For example process screens to display the process. Tags are used for transferring the data between PLC and HMI device in runtime. Similarly alarms are used for displaying operational states in runtime. To save the process value logs are used.

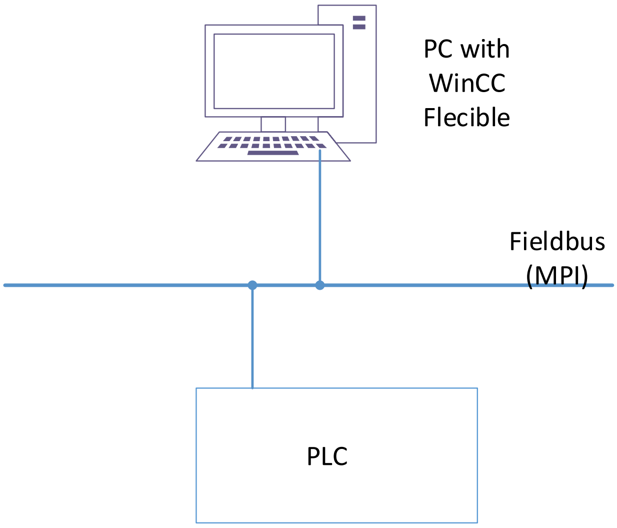


Figure 11 HMI connection with the PLC

The connection in the car track system is single-user system. If there would be more than one HMI connected in the process bus then it would be multiple- user system. The PC also works as a central function as it it-self does recipe management.

5.1 Establishment of communication

Figure below shows the configuration of parameters for the communication drivers.

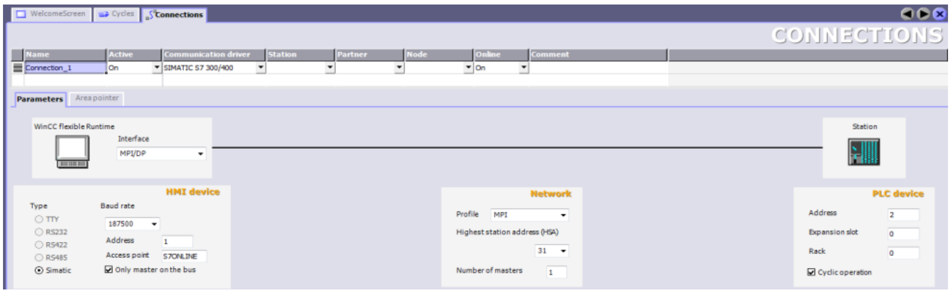


Figure 12 Connection configuration between WinCC flexible RT and PLC

In WinCC and automation systems (AS), to acquire the data the HMI sends request message to the automation system, then AS returns the requested data to the HMI in a response frame. Since the CPU used in the station is lowest version of S7, it only supports MPI communication. PPI, Profibus DP and TCI/IP (Profinet) is not possible to in these hardware criteria. The hardware components used for networking was PC adaptor. It converts RS 232 serial to RS-485 MPI. Instead using PC adaptor communication processer could also be used.

By using coordination area pointer detection of connection between PLC and HMI was done.

Table 7 Assignment of bits in the 'coordination area pointer

	High order byte								Low order byte							
	15							8	7					2	1	0
1st word	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	X
2nd word not assigned																

Life bit (bit 2)
Operation mode (bit 1)
Startup bit (bit 0)

HMI device inverts the life bit at interval of approximately one second. By querying this bit in PLC program, connection between PLC and HMI was possible.

5.2 Tags

Before linking HMI buttons, I/O or symbolic filed, tags should be created for external tags to define the memory location in the PLC. The tags created in HMI were created so that they match with applicable data types in PLC. All the tags were created with basic data types. By creating structures own data type creation is also possible. Few internal tags were created in order to communicate within HMI internally. These types of tags are stored in the memory of the HMI device.

5.3 Screens

The HMI was realized with 5 screens.

Welcome Screen

This screen is visible when at first when the runtime system is opened. It provides parameter selections for users in order to initiate the race. Users can also view the CPU mode weather it is at STOP mode or in RUN mode, if the CPU is at stop mode, it needs to be switched to RUN mode. Three difficulty level options are available. By selections of one option, other options are deselected. This was possible by using Invert bit and Reset other bits function in WinCC Flexible.

On the right side of the screen, options for race duration are available. User can either play unlimited laps or limited to certain number chosen by user. Third possibility is to choose certain time span. The time span is entered in the second's format. Cars are running until time is elapsed chosen by user. User can proceed to next screen only if two cars are placed at staring position.

Race screen

IO filed, text filed, Buttons, graphic filed etc. were used to input or output various variables.

Monitoring screen

This screen was made for monitoring and maintenance purpose. It contains information about states of sensors, car track mode and CPU mode.

6 CONCLUSION

Using PLC program, coded using STEP 7 technology from Siemens, the control of car track system has been succeeded. The speed of the car at every segment in the track changes accordingly. From the control panel, the process can be started or stopped. Other measurement variables can also be viewed from control panel at real time. The user can view the state of the race modes, alarms and other hardware component's status.

The speed was controlled by applying appropriate milliamp current through analog output module. The integer value was moved to peripheral output word address in order to supply the required milliamps value within the range of 0-20mA. The measurement of the speed of the car at every segment in the track has been partly successful.

The measurement of time duration between two events was done using standard function block. This was done in order to reduce programming length and increase simplicity. The thesis illustrated different ways of measuring time duration using STEP 7.

The researcher has become more proficient in logic design. The skills to design HMI were also achieved.

The difficult part was to know the output modules properties and way of supplying analog milliamps current to the track. Other difficulty during the execution of project was researcher had only basic of knowledge in SPEP 7 programming environment. The researcher had no knowledge of HMI programming.

SOURCES

Programming with STEP 7 – Manual from Siemens

Configuring Hardware and communication connections- Manual from Siemens

WinCC flexible 2008 Manual form Siemens

WinCC flexible 2008 Runtime- Manual from Siemens

Hamk.fi,

Moodle2.hamk.fi PLC course

Siemens application tool
http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&aktprim=0&siteid=cseus&lang=en&siteid=cseus&extranet=standard&viewreg=WW&groupid=4000003

Step 7 in seven steps; Books by C.T Jones
http://books.google.fi/books?id=iE_ZTO3P6NMC&printsec=frontcover#v=onepage&q&f=false

Automating with STEP 7 in STL and SCL: Hans Berger

Digital input modules (Manual chapter 3)

Analog Output Module SM 332 manual form Siemens

Options for communication via MPI
http://support.automation.siemens.com/WW/llisapi.dll/21537047?func=ll&objId=21537047&objAction=csView&nodeid0=33516848&lang=en&siteid=cseus&aktprim=0&extranet=standard&viewreg=WW&load=content&prodLstObjId=4067768&start=111&csSort=-DOCDATUMBEITRAG&subtype=133000

Options for recording time stamps
https://support.automation.siemens.com/WW/llisapi.dll?func=ll&objid=22818923&nodeid0=10805384&load=treecontent&lang=en&siteid=cseus&aktprim=0&objaction=csview&extranet=standard&viewreg=WW#A643077

How to connect S7 300 and WinCC flexible with PRFIBUS or MPI
Electronic document at Siemens website

APPENDICES

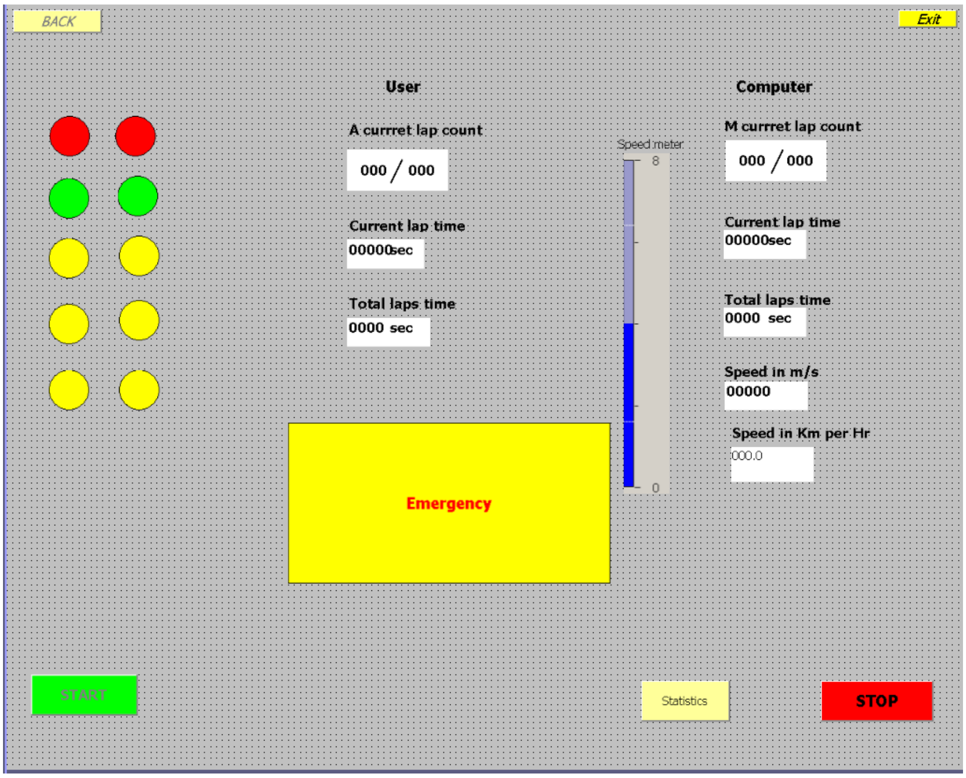


Figure 13 Race screen

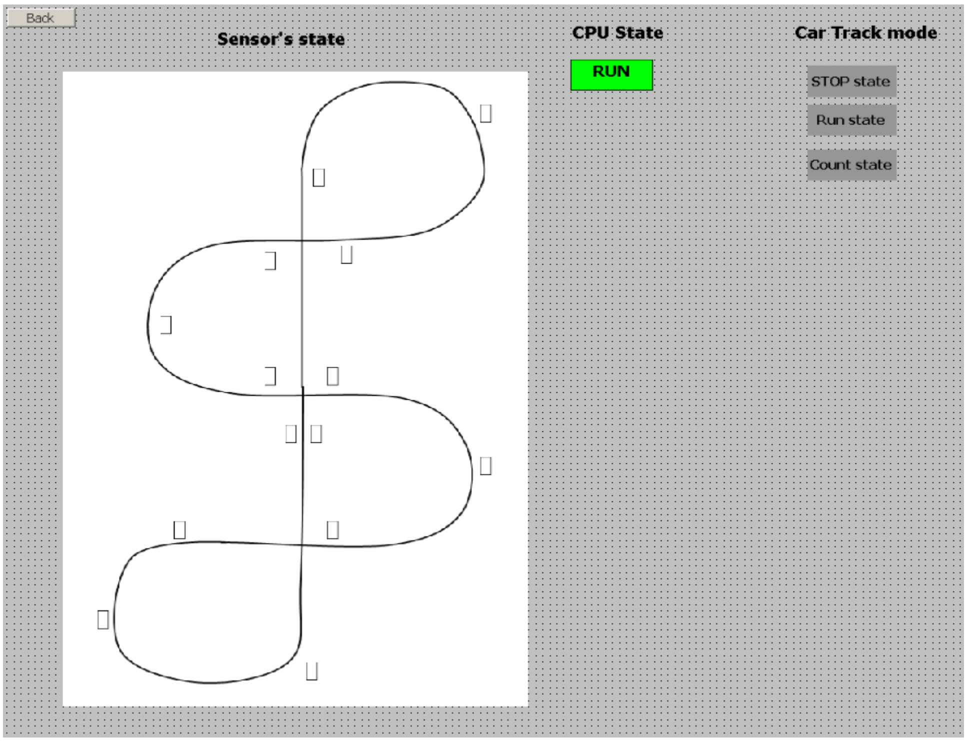


Figure 14 Monitoring screen

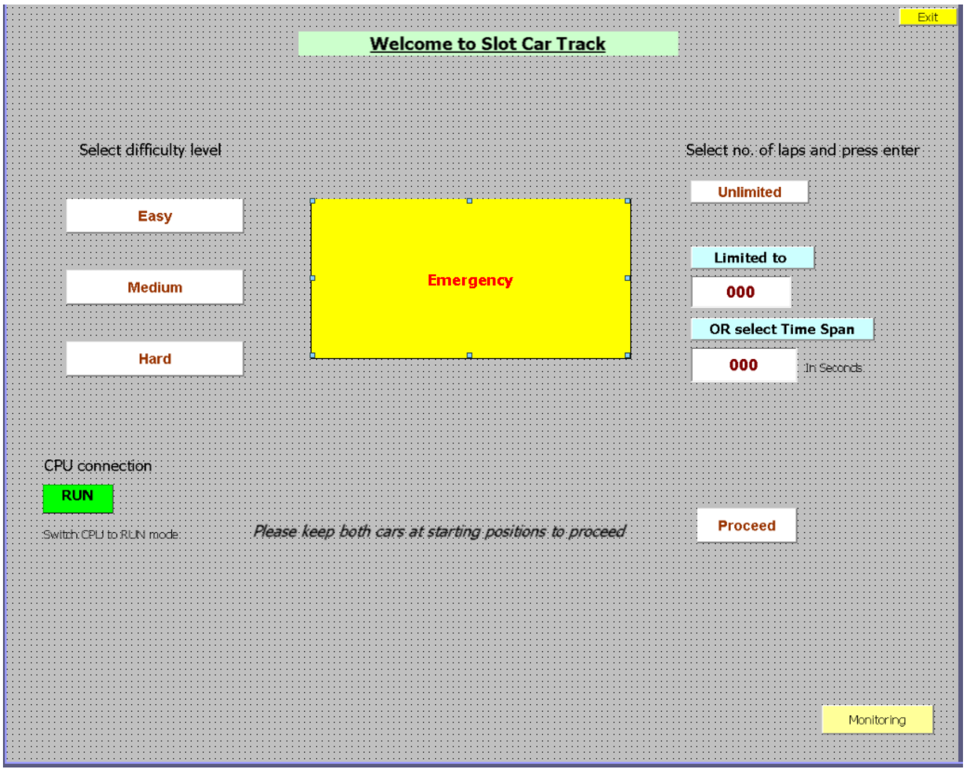


Figure 15 Welcome screen

Symbol Editor - [S7 Program (Symbols) -- PGRPJ\SIMATIC 300\CPU 312IFM]

Symbol Table Edit Insert View Options Window Help

All Symbols

	Statu	Symbol	Address		Data type		Comment
1		A all laps	M	4.1	BOOL		
2		A current lap cou...	C	1	COUNTER		
3		A sensor	I	2.7	BOOL		Start position automated side
4		AT1	MD	60	TIME		
5		COMPLETE REST...	OB	100	OB	100	Complete Restart
6		Count mode	M	0.1	BOOL		Counting to begin the race
7		Count state com...	M	4.0	BOOL		
8		Count_f	FC	2	FC	2	
9		Curv1.0	I	3.7	BOOL		
1		Curv1.1	I	1.7	BOOL		
1		Curv1.2	I	1.6	BOOL		
1		Curv2.0	I	3.5	BOOL		
1		Curv2.1	I	1.5	BOOL		
1		Curv2.2	I	1.4	BOOL		
1		Curv3.0	I	3.3	BOOL		
1		Curv3.1	I	1.3	BOOL		
1		Curv3.2	I	1.2	BOOL		
1		Curv4.0	I	3.1	BOOL		
1		Curv4.1	I	1.1	BOOL		
2		Curv4.2	I	1.0	BOOL		
2		E. Stop	I	124.0	BOOL		Emergency stop
2		Easy	FC	4	FC	4	
2		Green led	Q	124.2	BOOL		
2		Hard	FC	6	FC	6	
2		HMI_data	DB	1	DB	1	
2		Lap counter	FC	7	FC	7	
2		M All laps	M	4.2	BOOL		
2		M current lap co...	C	2	COUNTER		
2		M sensor	I	0.7	BOOL		Start position manual side
3		Manual car	FC	8	FC	8	
3		Medium	FC	5	FC	5	
3		MT1	MD	68	TIME		
3		Output 1	Q	124.0	BOOL		
3		PE_run mode	M	31.1	BOOL		
3		Race_duration ti...	T	4	TIMER		
3		Red led	Q	124.3	BOOL		
3		Run mode	M	0.2	BOOL		
3		Run_f	FC	3	FC	3	
3		Speed_calc	FC	11	FC	11	
4		Speed_call	FC	10	FC	10	
4		Stop_f	FC	1	FC	1	
4		Time calculation	FC	9	FC	9	
4		time PE runmode	MD	72	TIME		
4		Time run mode ON	MD	64	TIME		
4		TIME_TCK	SFC	64	SFC	64	Read the System Time
4		Wait for start	M	0.0	BOOL		
4		Voltage out	PQW	272	WORD		
4		Yellow low	Q	124.5	BOOL		
4		Yellow mid	Q	124.4	BOOL		
5		Yellow top	Q	124.1	BOOL		

Press F1 to get Help.

Instructions to the user

Instructions

Power supply to the CPU and PC

Turn on PC, supply power to S7 300 station

Switch CPU to the RUN mode

Green led should be on.

Open file "Car track system.RT" form Desktop

The welcome screen should open

Be sure emergency push bottom is not pressed

If button is pressed, yellow window flashes in the runtime system

Select difficulty level from the left

Click to select

Select run mode options from the right

Choose one options and enter the value asked, press enter after typing

Place both cars at starting positions

Options to proceed to next screen appears when placing both cars at starting position

Now press START

Function block diagrams

SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\OB1 - <offline> 05/08/2013 12:42:45 PM

OB1 - <offline>

""

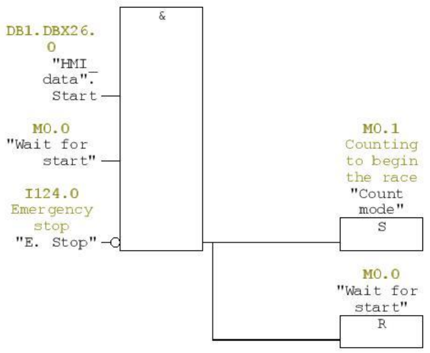
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/07/2013 01:02:18 PM
Interface: 04/02/2013 03:32:08 PM
Lengths (block/logic/data): 00366 00222 00026

Name	Data Type	Address	Comment
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started
time_elapsed	Bool	20.0	
times5	S5Time	22.0	

Block: OB1 "Main Program Sweep (Cycle)"

Main sequence : Stop stst > count state > run state
And calling all other user porgrammed blocks

Network: 1 Counting to begin the race



OB1 - <offline>

""

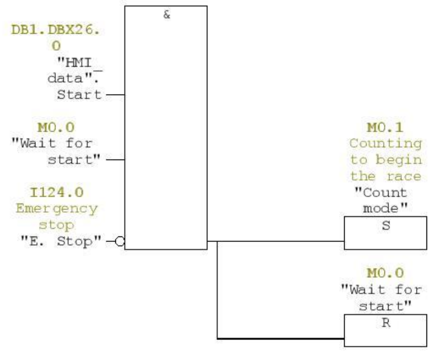
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/07/2013 01:02:18 PM
Interface: 04/02/2013 03:32:08 PM
Lengths (block/logic/data): 00366 00222 00026

Name	Data Type	Address	Comment
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started
time_elapsed	Bool	20.0	
times5	S5Time	22.0	

Block: OB1 "Main Program Sweep (Cycle)"

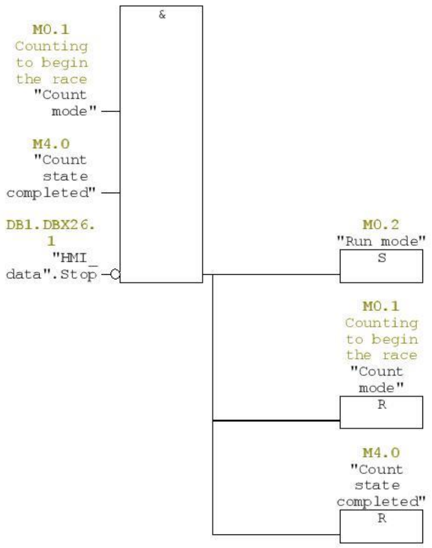
Main sequence : Stop stst > count state > run state
And calling all other user porgrammed blocks

Network: 1 Counting to begin the race

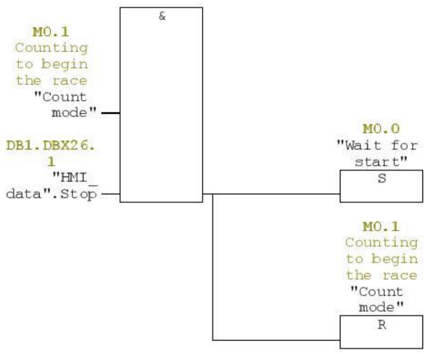


SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\OB1 - <offline> 05/08/2013 12:42:45 PM

Network: 2

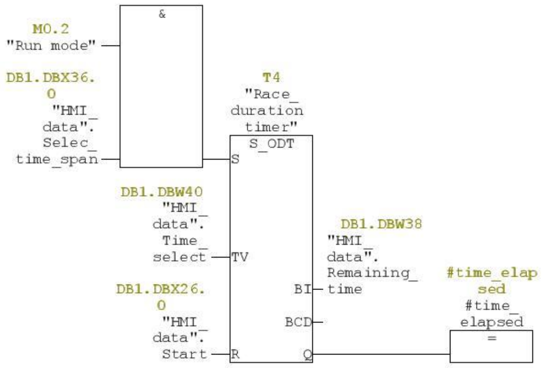


Network: 3

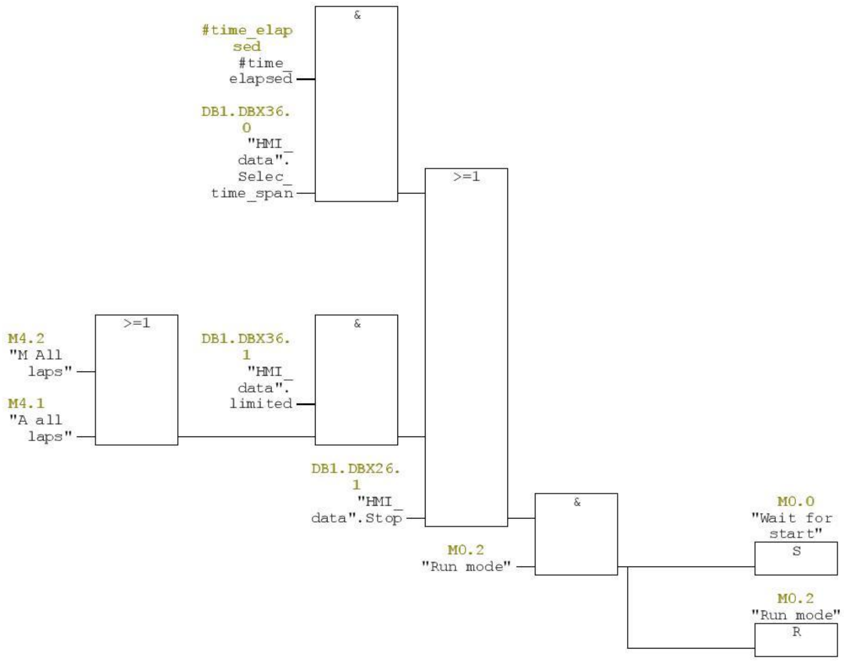


Network: 4

Setting time value for rum mode duration.



Network: 5



SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\OB1 - <offline> 05/08/2013 12:42:46 PM

Network: 6

FC1
"Stop_f"
EN ENO

Network: 7

FC2
"Count_f"
EN ENO

Network: 8

FC3
"Run_f"
EN ENO

Network: 9

FC7
"Lap counter"
EN ENO

Network: 10

FC9
"Time calculation"
EN ENO

Network: 11

FC10
"Speed_call"
EN ENO

Page 4 of 4

SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\OB100 - <offline> 05/08/2013 12:49:35 PM

OB100 - <offline>

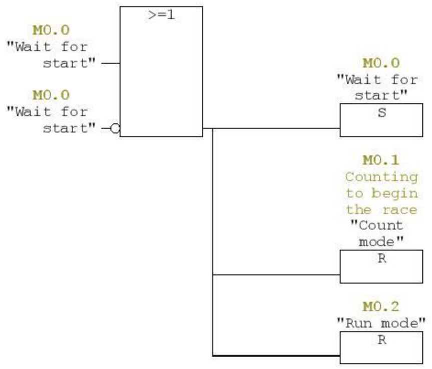
"COMPLETE RESTART" Complete Restart
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/07/2013 12:45:34 PM
Interface: 02/15/1996 04:51:10 PM
Lengths (block/logic/data): 00122 00012 00020

Name	Data Type	Address	Comment
TEMP		0.0	
OB100_EV_CLASS	Byte	0.0	16#13, Event class 1, Entering event state, Event logged in diagnostic buffer
OB100_STRTUP	Byte	1.0	16#81/82/83/84 Method of startup
OB100_PRIORITY	Byte	2.0	Priority of OB Execution
OB100_OB_NUMBR	Byte	3.0	100 (Organization block 100, OB100)
OB100_RESERVED_1	Byte	4.0	Reserved for system
OB100_RESERVED_2	Byte	5.0	Reserved for system
OB100_STOP	Word	6.0	Event that caused CPU to stop (16#4xxx)
OB100_STRT_INFO	DWord	8.0	Information on how system started
OB100_DATE_TIME	Date_And_Time	12.0	Date and time OB100 started

Block: OB100 "Complete Restart"

Wait for start is active on restart and startup of CPU. Reset ohter states

Network: 1



Page 1 of 1

FC1 - <offline>

"Stop_f"
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/08/2013 11:50:00 AM
Interface: 03/17/2013 11:47:54 AM
Lengths (block/logic/data): 00130 00032 00000

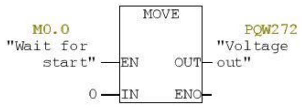
Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC1

Car track in the stop state, zero voltage in the track(move zero to PQW 272);
red led is on at the display.

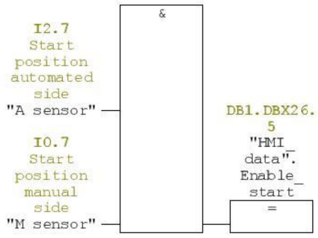
Network: 1

Network: 2



Network: 3

sensors at starting position should be high, this enable start option in HMI



SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\FC1 - <offline> 05/08/2013 12:49:50 PM

FC1 - <offline>

"Stop_f"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/08/2013 11:50:00 AM
Interface: 03/17/2013 11:47:54 AM
Lengths (block/logic/data): 00130 00032 00000

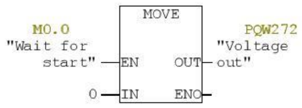
Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC1

Car track in the stop state, zero voltage in the track(move zero to PQW 272);
red led is on at the display.

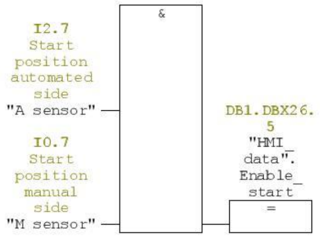
Network: 1

Network: 2



Network: 3

sensors at starting position should be high, this enable start option in HMI



Page 1 of 2

FC2 - <offline>

"Count_f"

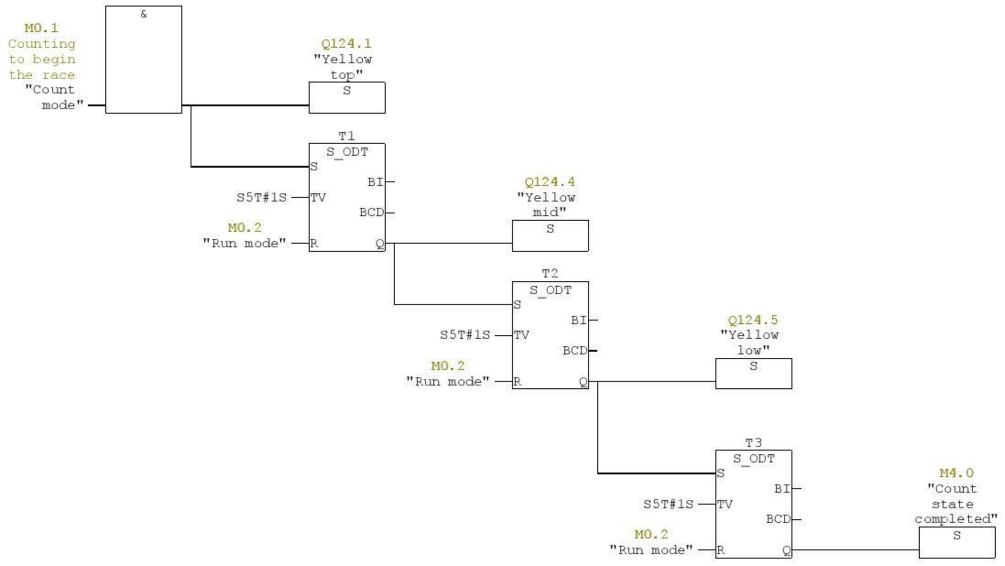
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/07/2013 12:49:21 PM
Interface: 03/17/2013 11:49:04 AM
Lengths (block/logic/data): 00210 00116 00002

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC2

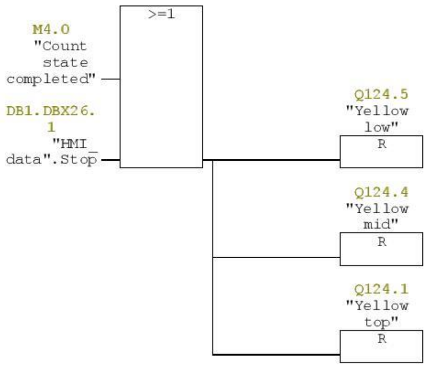
Three led ON simultaneously, three times after delay of 1s; all led reset after count mode completed

Network: 1



SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\FC2 - <offline> 05/08/2013 12:50:03 PM

Network: 2



Page 2 of 2

SIMATIC   PGRPJ\SIMATIC 300\CPU 312IFM\...\FC3 - <offline>   05/08/2013 12:50:17 PM

FC3 - <offline>

"Run_f"

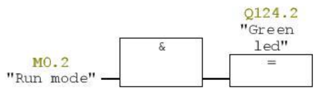
Name:		**Family:**	
Author:		**Version:**	0.1
		Block version:	2
Time stamp Code:			05/07/2013 12:50:55 PM
	Interface:		03/17/2013 11:49:21 AM
Lengths (block/logic/data):			00196 00090 00002

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

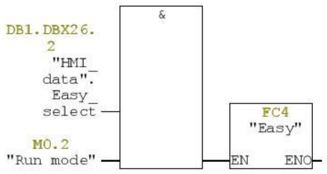
Block: FC3

Calling three blocks which defines voltage output to track at correnponding difficulty level slection. Green led is on when car track is at run state.

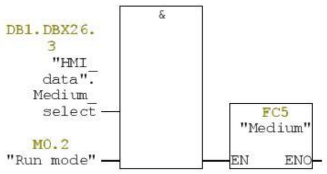
Network: 1



Network: 2



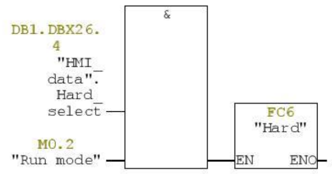
Network: 3



Page 1 of 2

SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\FC3 - <offline> 05/08/2013 12:50:17 PM

Network: 4



Page 2 of 2

FC4 - <offline>

"Easy"

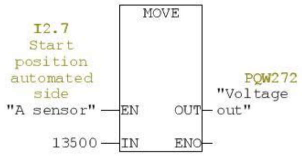
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/07/2013 12:51:56 PM
Interface: 03/17/2013 11:49:37 AM
Lengths (block/logic/data): 00352 00210 00000

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

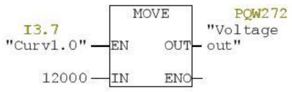
Block: FC4

Moving interger values in correspondence to the position of car

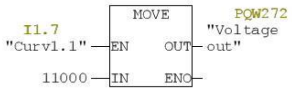
Network: 1



Network: 2



Network: 3



Network: 4

```
                   MOVE        PQW272
   I1.6                        "Voltage
"Curv1.2" ──EN          OUT── out"

   13500 ──IN           ENO──
```

Network: 5

```
                   MOVE        PQW272
   I3.5                        "Voltage
"Curv2.0" ──EN          OUT── out"

   12500 ──IN           ENO──
```

Network: 6

```
                   MOVE        PQW272
   I1.5                        "Voltage
"Curv2.1" ──EN          OUT── out"

   11500 ──IN           ENO──
```

Network: 7

```
                   MOVE        PQW272
   I1.4                        "Voltage
"Curv2.2" ──EN          OUT── out"

   13500 ──IN           ENO──
```

Network: 8

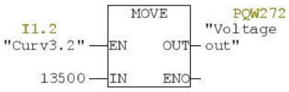
```
                   MOVE        PQW272
   I3.3                        "Voltage
"Curv3.0" ──EN          OUT── out"

   12500 ──IN           ENO──
```

Network: 9

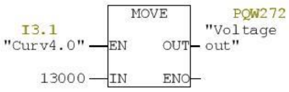
```
                   MOVE        PQW272
   I1.3                        "Voltage
"Curv3.1" ──EN          OUT── out"

   11500 ──IN           ENO──
```

SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\FC4 - <offline> 05/08/2013 12:50:30 PM

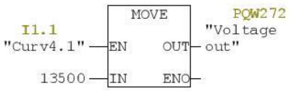
Network: 10



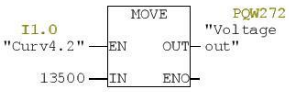
Network: 11



Network: 12



Network: 13



FC5 - <offline>

"Medium"

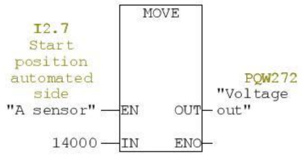
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/07/2013 12:52:05 PM
Interface: 03/17/2013 11:52:17 AM
Lengths (block/logic/data): 00352 00210 00000

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

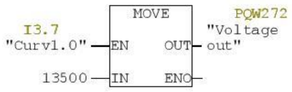
Block: FC5

Moving interger values in correspondence to the position of car

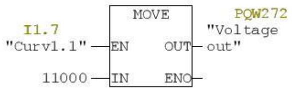
Network: 1



Network: 2

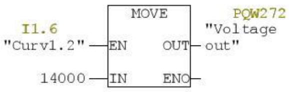


Network: 3

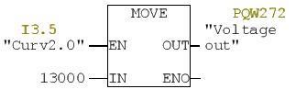


SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\FC5 - <offline> 05/08/2013 12:50:43 PM

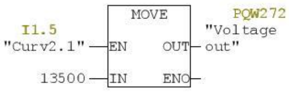
Network: 4



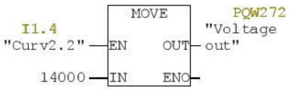
Network: 5



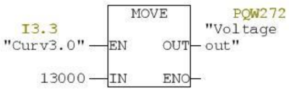
Network: 6



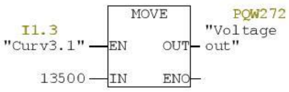
Network: 7



Network: 8



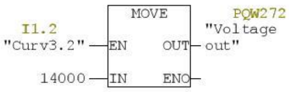
Network: 9



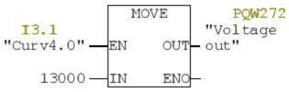
Page 2 of 3

SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\FC5 - <offline> 05/08/2013 12:50:43 PM

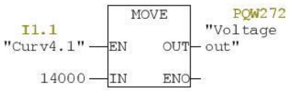
Network: 10



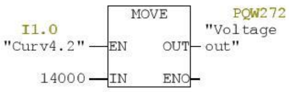
Network: 11



Network: 12



Network: 13



Page 3 of 3

SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\FC6 - <offline> 05/08/2013 12:50:53 PM

FC6 - <offline>

"Hard"

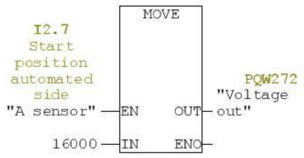
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/07/2013 12:52:15 PM
Interface: 03/17/2013 11:52:32 AM
Lengths (block/logic/data): 00346 00204 00000

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

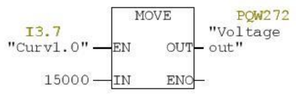
Block: FC6

Moving interger values in correspondence to the position of car

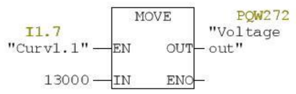
Network: 1



Network: 2

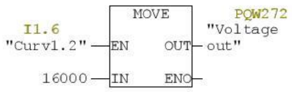


Network: 3

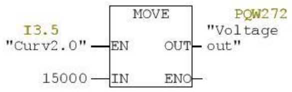


SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\FC6 - <offline> 05/08/2013 12:50:54 PM

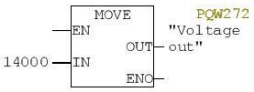
Network: 4



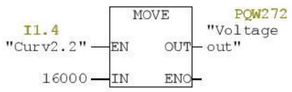
Network: 5



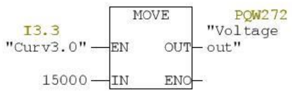
Network: 6



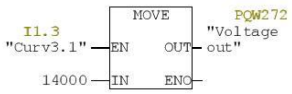
Network: 7



Network: 8

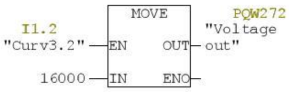


Network: 9

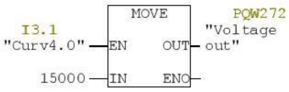


SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\FC6 - <offline> 05/08/2013 12:50:54 PM

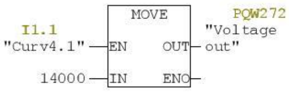
Network: 10



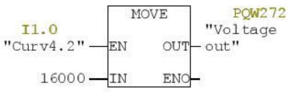
Network: 11



Network: 12



Network: 13



SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\FC7 - <offline> 05/08/2013 12:51:05 PM

FC7 - <offline>

"Lap counter"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/07/2013 12:55:33 PM
Interface: 03/18/2013 10:47:50 AM
Lengths (block/logic/data): 00238 00118 00008

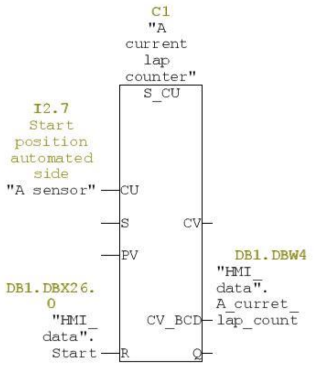
Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
TEMP0	Int	0.0	
TEMP1	Int	2.0	
TEMP2	Int	4.0	
Aall	Word	6.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC7

UP counter counts number of rising edges of 'A sesnor' and 'M sensor'. Current count value is continiously compared with the value in HMI.

Network: 1

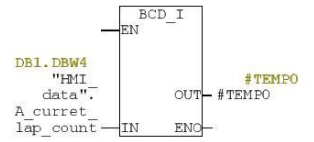
For automated car



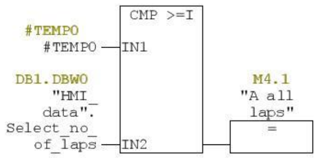
Page 1 of 3

SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\FC7 - <offline> 05/08/2013 12:51:05 PM

Network: 2

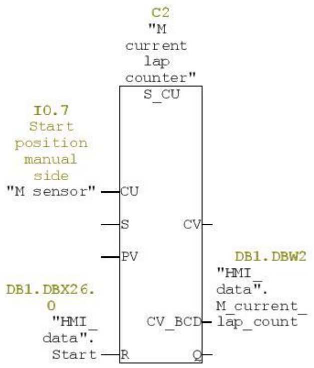


Network: 3

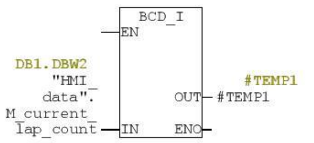


Network: 4

For manual car



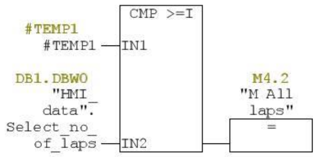
Network: 5



Page 2 of 3

SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\FC7 - <offline> 05/08/2013 12:51:05 PM

Network: 6



Page 3 of 3

FC9 - <offline>

"Time calculation"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/08/2013 12:13:11 PM
Interface: 03/27/2013 10:14:17 AM
Lengths (block/logic/data): 00376 00240 00030

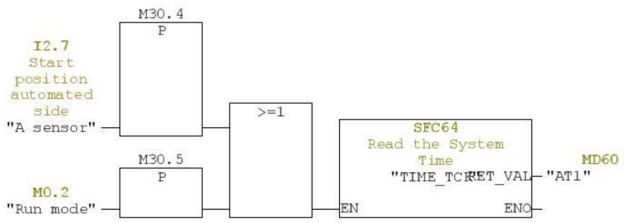
Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
TIME1	Time	0.0	A single lap initiation
TIME2	Time	4.0	Time until run mode is active
TIME3	Time	8.0	M single lap initiation
TIME4	Time	12.0	run mode initiation
TIME5	Time	16.0	
TIME6	Time	20.0	
TIME7	Time	24.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC9

Race duration measurement

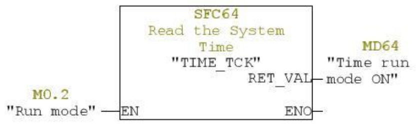
Network: 1

Saving time stamp of CPU at rising edge of rumnode or automated car car leaves starting position



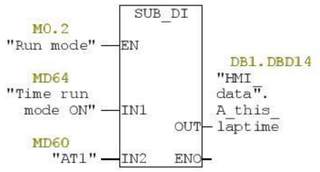
Network: 2

Saving continuously stamp time of CPU when car is at run mode



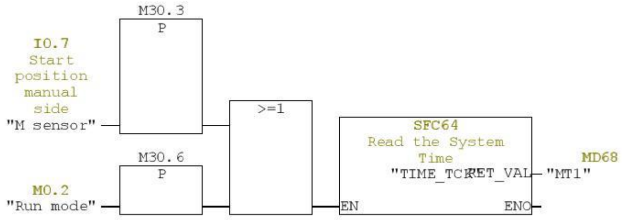
Network: 3

Difference between two time values

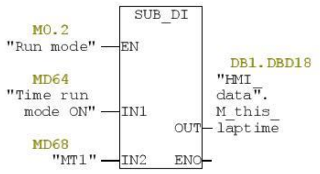


Network: 4

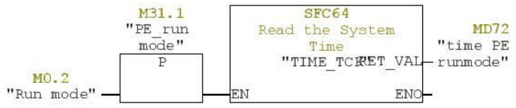
same process for manaual car



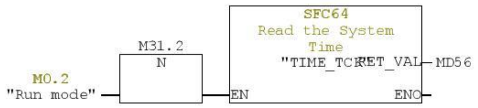
Network: 5



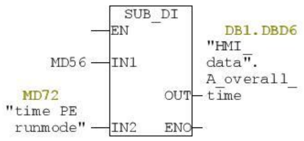
Network: 6



Network: 7



Network: 8


SUB_DI
EN
MD56
IN1
MD72
"time PE
runmode"
IN2
OUT
ENO
DB1.DBD6
"HMI_
data".
A_overall_
time

SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\FC10 - <offline> 05/08/2013 12:52:21 PM

FC10 - <offline>

"Speed_call"

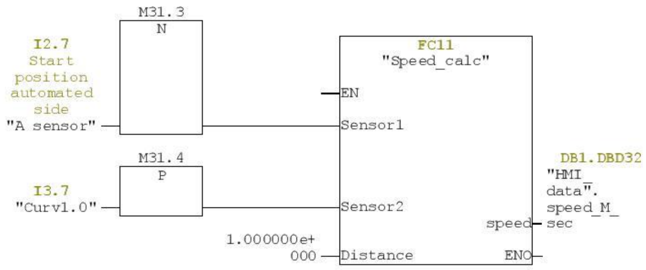
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/07/2013 12:57:36 PM
Interface: 03/28/2013 09:47:25 AM
Lengths (block/logic/data): 01110 00990 00010

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

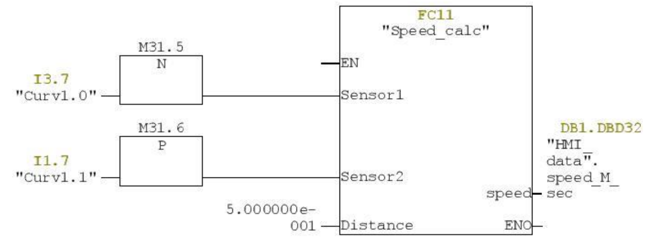
Block: FC10

Calling Speed_calc block

Network: 1



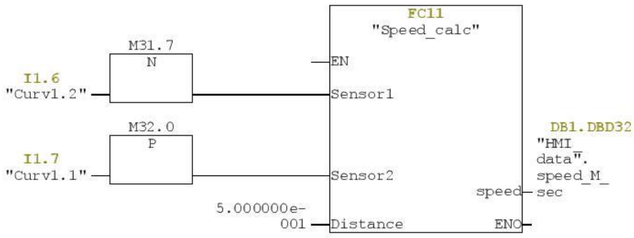
Network: 2



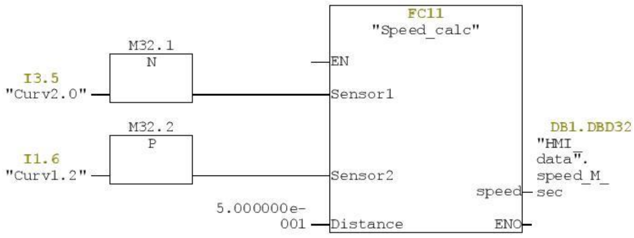
Page 1 of 4

SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\FC10 - <offline> 05/08/2013 12:52:21 PM

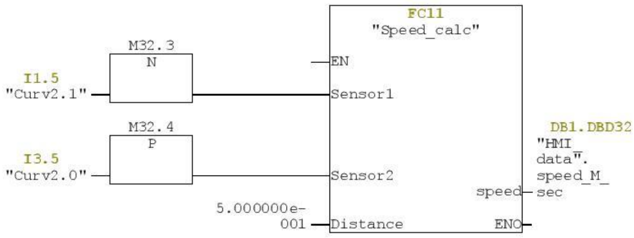
Network: 3



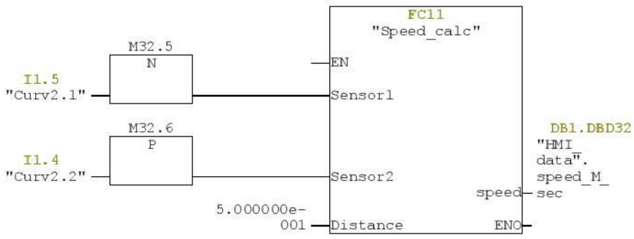
Network: 4



Network: 5

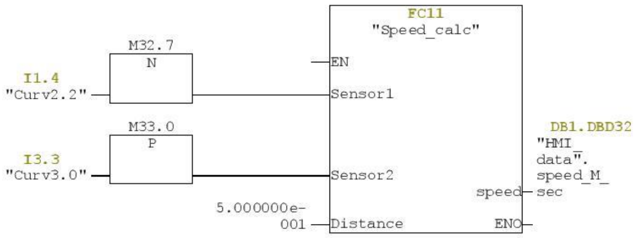


Network: 6

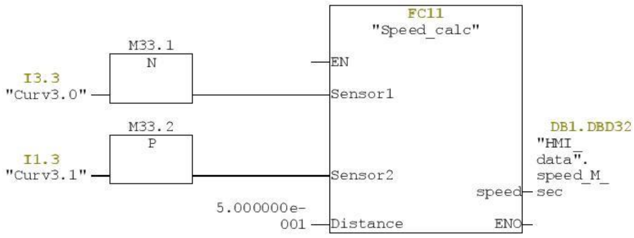


Page 2 of 4

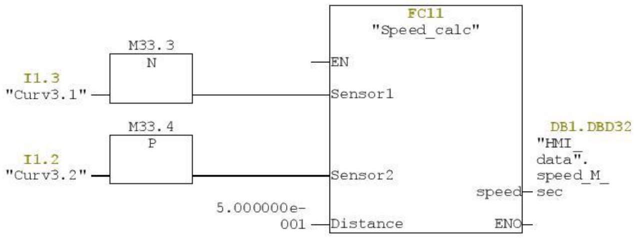
Network: 7



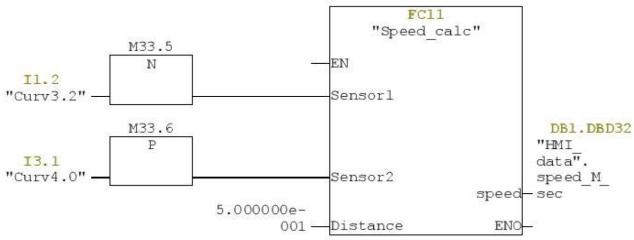
Network: 8



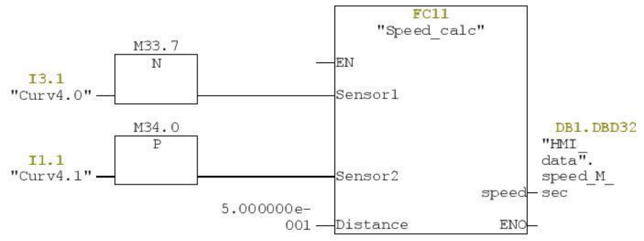
Network: 9



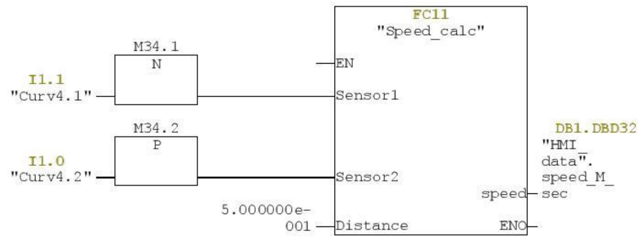
Network: 10



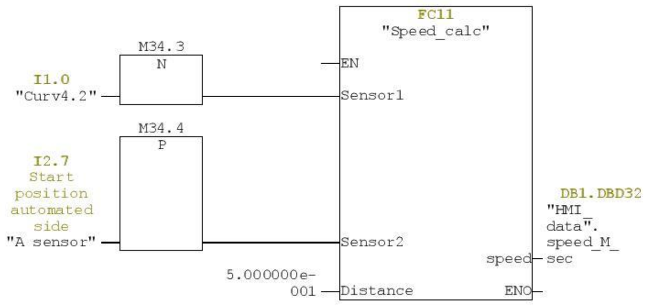
Network: 11



Network: 12



Network: 13



Network: 14

FC11 - <offline>

"Speed_calc"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/08/2013 12:13:57 PM
Interface: 05/06/2013 10:35:23 AM
Lengths (block/logic/data): 00304 00170 00032

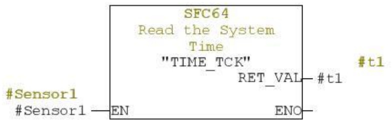
Name	Data Type	Address	Comment
IN		0.0	
Sensor1	Bool	0.0	
Sensor2	Bool	0.1	
Distance	Real	2.0	
OUT		0.0	
speed	Real	6.0	
IN_OUT		0.0	
TEMP		0.0	
meter	Int	0.0	
t2	Time	2.0	
timeinsec	Real	6.0	
t_real	Real	10.0	
t1	Time	14.0	
t2_t1	DInt	18.0	
mpersec	Real	22.0	
commonsesortime	Time	26.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC11

Speed calculation at each segment in the car track

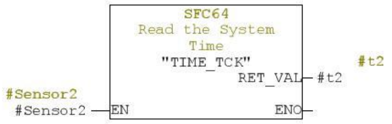
Network: 1

Time at first sensor



Network: 2

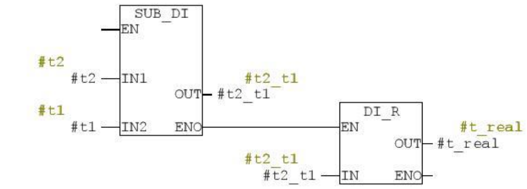
Time at second sensor



SIMATIC PGRPJ\SIMATIC 300\CPU 312IFM\...\FC11 - <offline> 05/08/2013 12:53:32 PM

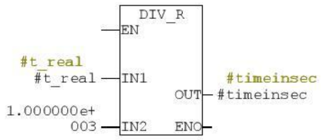
Network: 3

Time difference and conversion to real data type



Network: 4

Time in Seconds



Network: 5

Speed = distance/time taken