11. FLOWCHART BASED DESIGN

11.1 INTRODUCTION

A flowchart is ideal for a process that has sequential process steps. The steps will be executed in a simple order that may change as the result of some simple decisions. The symbols used for flowcharts are shown in Figure 11.1. These blocks are connected using arrows to indicate the sequence of the steps. The different blocks imply different types of program actions. Programs always need a **start** block, but PLC programs rarely stop so the **stop** block is rarely used. Other important blocks include **operations** and **decisions**. The other functions may be used but are not necessary for most PLC applications.

Figure 11.1 Flowchart Symbols

**Topics:**
- Objectives:
  - Able to describe a process with a flowchart.
  - Be able to convert a flowchart to ladder logic.

**Describing process control using flowcharts**

**Conversion of flowcharts to ladder logic**
A flowchart is shown in Figure 11.2 for a control system for a large water tank. When a start button is pushed the tank will start to fill, and the flow out will be stopped. When full, or the stop button is pushed the outlet will open up, and the flow in will be stopped. In the flowchart the general flow of execution starts at the top. The first operation is to open the outlet valve and close the inlet valve. Next, a single decision block is used to wait for a button to be pushed. When the button is pushed the yes branch is followed and the inlet valve is opened, and the outlet valve is closed. Then the flow chart goes into a loop that uses two decision blocks to wait until the tank is full, or the stop button is pushed. If either case occurs the inlet valve is closed and the outlet valve is opened. The system then goes back to wait for the start button to be pushed again. When the controller is on the program should always be running, so only a start block is needed. Many beginners will neglect to put in checks for stop buttons.
The general method for constructing flowcharts is:

1. Understand the process.
2. Determine the major actions, these are drawn as blocks.
3. Determine the sequences of operations, these are drawn with arrows.

Figure 11.2
A Flowchart for a Tank Filler

START
Open outlet valve
start button pushed?
Open inlet valve
Close outlet valve
Is tank full?
stop button pushed?
Open outlet valve
Close inlet valve
no
no
yes
yes

Close inlet valve
4. When the sequence may change use decision blocks for branching.

Once a flowchart has been created ladder logic can be written. There are two basic techniques that can be used, the first presented uses blocks of ladder logic code. The second uses normal ladder logic.

11.2 BLOCK LOGIC

The first step is to name each block in the flowchart, as shown in Figure 11.3. Each of the numbered steps will then be converted to ladder logic.
Figure 11.3
Labeling Blocks in the Flowchart

Each block in the flowchart will be converted to a block of ladder logic. To do this we will use the MCR (Master Control Relay) instruction (it will be discussed in more detail later.) The instruction is shown in Figure 11.4, and will appear as a matched pair of outputs labelled **MCR**. If the first MCR line is true then the ladder logic on the following lines will be scanned as normal to the second MCR. If the first line is false the lines to the START.

Open outlet valve
start button pushed?
Open inlet valve
Close outlet valve
Is tank full?
stop button pushed?
Open outlet valve
Close inlet valve

no
no
yes
yes
The first part of the ladder logic required will reset the logic to an initial condition, as shown in Figure 11.5. The line will only be true for the first scan of the PLC, and at that time it will turn on the flowchart block \( F1 \) which is the reset all values off operation. All other operations will be turned off.

If \( A \) is true then the MCR will cause the ladder in between to be executed. If \( A \) is false the outputs are forced off.

Note: We will use MCR instructions to implement some of the state based programs. This allows us to switch off part of the ladder logic. The one significant note to remember is that any normal outputs (not latches and timers) will be FORCED OFF. Unless this is what you want, put the normal outputs outside MCR blocks.
The ladder logic for the first state is shown in Figure 11.6. When \( F1 \) is true the logic between the MCR lines will be scanned, if \( F1 \) is false the logic will be ignored. This logic turns on the outlet valve and turns off the inlet valve. It then turns off operation \( F1 \), and turns on the next operation \( F2 \).
The ladder logic for operation \( F_1 \) is simple, and when the start button is pushed, it will turn off \( F_2 \) and turn on \( F_3 \). The ladder logic for operation \( F_3 \) opens the inlet valve and moves to operation \( F_4 \).

**STEP 3:** Write ladder logic for each function in the flowchart.
Figure 11.7 Ladder Logic for Flowchart Operations

The ladder logic for operation $F_4$ turns off $F_4$, and if the tank is full it turns on $F_6$, otherwise $F_5$ is turned on. The ladder logic for operation $F_5$ is very similar.
The ladder logic for operation \( F_6 \) turns the outlet valve on and turns off the inlet valve. It then ends operation \( F_6 \) and returns to operation \( F_2 \).
In general there is a preference for methods that do not use MCR statements or latches. The flowchart used in the previous example can be implemented without these instructions using the following method. The first step to this process is shown in Figure 11.10. As before each of the blocks in the flowchart are labelled, but now the connecting arrows (transitions) in the diagram must also be labelled. These transitions indicate when another function block will be activated.
The first section of ladder logic is shown in Figure 11.11. This indicates when the transitions between functions should occur. All of the logic for the transitions should be kept together, and appear before the state logic that follows in Figure 11.12.

1. Open outlet valve
2. Start button pushed?
3. Open inlet valve
4. Close outlet valve
5. Is tank full?
6. Stop button pushed?
7. Open outlet valve
8. Close inlet valve
9. no
10. yes
11. no
12. yes

The diagram shows the flowchart blocks and arrows.
The logic shown in Figure 11.12 will keep a function on, or switch to the next function. Consider the first ladder rung for $F_1$, it will be turned on by transition $T_1$ and once function $F_1$ is on it will keep itself on, unless $T_2$ occurs shutting it off. If $T_2$ has occurred the next line of ladder logic will turn on $F_2$. The function logic is followed by output logic that relates output values to the active functions.
Figure 11.12
The Function Logic and Outputs

F1 T1 T2 F1
F2 T2 T3 F2
F3 T3 T4 F3
F4 T4 T5 F4
F5 T5 T4 F5
F6 T6 T2 F6
F1 outlet F2 F6
F3 inlet F4 F5
11.15 PRACTICE PROBLEMS

1. Convert the following flow chart to ladder logic.

2. Draw a flow chart for cutting the grass, then develop ladder logic for three of the actions/decisions.

3. Design a garage door controller using a flowchart. The behavior of the garage door controller is as follows,
   - There is a single button in the garage, and a single button remote control.
   - When the button is pushed the door will move up or down.
   - If the button is pushed once while moving, the door will stop, a second push will start motion again in the opposite direction.
   - There are top/bottom limit switches to stop the motion of the door.
   - There is a light beam across the bottom of the door. If the beam is cut while the door is closing the door will stop and reverse.
   - There is a garage light that will be on for 5 minutes after the door opens or closes.
2. Start
Get mower and Fill mower
Is gas can empty?
Pull cord
Is all lawn cut?
Push mower
Stop mower
Put gas and get gas mower away

F1 F2 F3 F4 F5 F6 F7 F8 F9 F10
PLC Flowchart - 11.19

MCR
F3
MCR
F4
L
U
F4
MCR
MCR
F5
L
U
T4:0/DN
pour gas
T4:0/DN
fill gas tank
gas can full
TON
Timer T4:0
Delay 5s
F5
MCR
MCR
F6
L
U
F5
mower on
F6
mower on
pull cord
cord pulled
cord pulled
F6
ETC.....................
3. ST 1
   is remote or button pushed?
   yes
   no
   turn on door close
   is light beam on?
   no
   yes
   limit pushed?
   turn off door close
   is remote or button pushed?
   yes
   turn on door open
   is remote or button or top limit pushed?
   yes
   limit pushed?
   turn off door open
PLC Flowchart - 11.21

First scan

ST 1

ST 2

ST 3

ST 4

ST 5

ST 6

ST 7

ST 8

ST 9

T4:0 preset 300s

T4:0/DN garage light

Door open

Door close
PLC Flowchart

MCR

ST 5

MCR

U

L

ST 5

ST 6

MCR

ST 6

MCR

U

L

ST 6

ST 7

MCR

ST 7

MCR

U

L

ST 7

ST 8

L

Door close

MCR

ST 8

MCR

U

L

ST 8

Door open
1. Develop ladder logic for the flowchart below.
2. Use a flow chart to design a parking gate controller.

Start

Turn A on

Is B on?

Turn A off

Is C on?

- keycard entry
- gate
- cars enter/leave
- the gate will be raised by one output and lowered by another. If the gate gets stuck an over current detector will make a PLC input true. If this is the case the gate should reverse and the light should be turned on indefinitely.
- if a valid keycard is entered a PLC input will be true. The gate is to rise and stay open for 10 seconds.
- when a car is over the car detector a PLC input will go true. The gate is to open while this detector is active. If it is active for more than 30 seconds the light should also turn on until the gate closes.
3. A welding station is controlled by a PLC. On the outside is a safety cage that must be closed while the cell is active. A belt moves the parts into the welding station and back out. An inductive proximity sensor detects when a part is in place for welding, and the belt is stopped. To weld, an actuator is turned on for 3 seconds. As normal, the cell has start and stop push buttons.

a) Draw a flow chart
b) Implement the chart in ladder logic

4. Convert the following flowchart to ladder logic.

5. A machine is being designed to wrap boxes of chocolate. The boxes arrive at the machine on a conveyor belt. The list below shows the process steps in sequence.

1. The box arrives and is detected by an optical sensor (P), after this the conveyor is stopped (C) and the box is clamped in place (H).
2. A wrapping mechanism (W) is turned on for 2 seconds.
3. A sticker cylinder (S) is turned on for 1 second to put consumer labelling on the box.

Inputs
- DOOR OPEN (NC)
- START (NO)
- STOP (NC)
- PART PRESENT

Outputs
- CONVEYOR ON
- WELD
3. The conveyor (C) is turned on.
4. The clamp (H) is turned off and the conveyor (C) is turned on.
5. After the box leaves the system returns to an idle state.

Develop ladder logic for the system using a flowchart. Don't forget to include regular start and stop inputs.