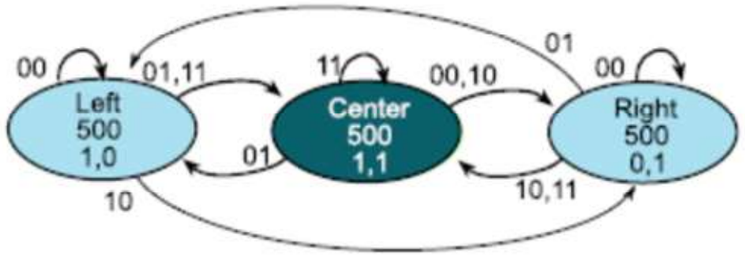
SOURCE MATERIAL
USE LineFollowFSMmain.c

Lab 7: Finite State Machine



1) The FSM in Figure 2 gets confused (has a bug) if the robot is off little bit to the left (input is 01, and the machine is oscillating between the Left and Center states) and then goes completely off the line to the left (input is 00). In this machine, if it happens to be in the Center state when it goes off the line, it will incorrectly move to the Right state even though the robot went off to the left. You will solve this problem by implementing two left states (so it oscillates between the two left states when a little left). For symmetry, you will implement two right states as well. Figure 4 shows a partial solution. If the input is 11, then the output should remain 11. If the input goes to 01 (it is a little left), then the output should toggle 1,0↔1,1 causing a slight right turn. Similarly, if the input goes to 10 (it is a little right), then the output should toggle 0,1↔1,1 causing a slight left turn.

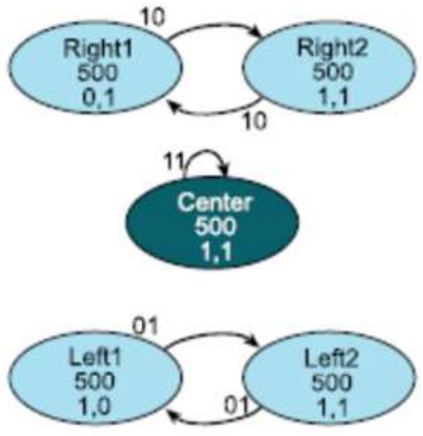
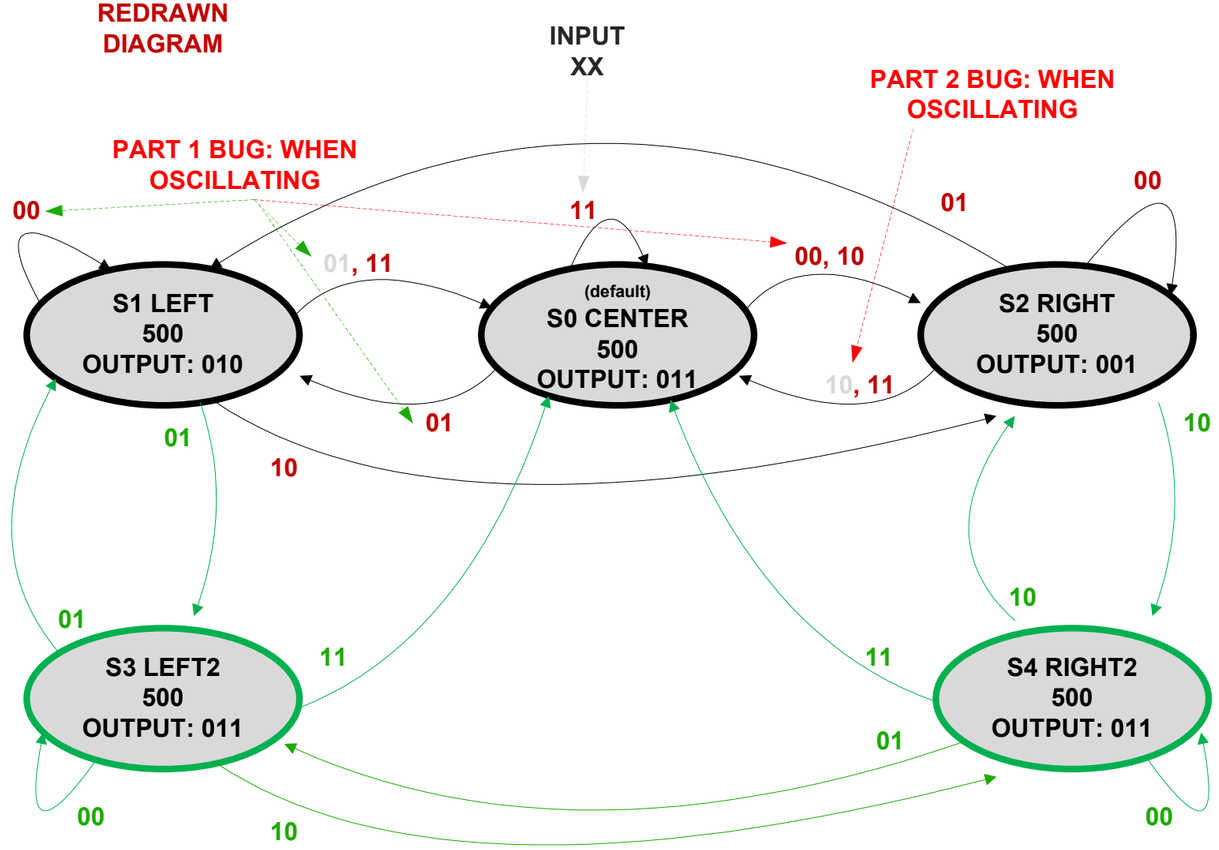


Figure 4. Expanded FSM state graph. The time in each state is shown in 1ms units.

REDRAWN DIAGRAM



MODIFICATIONS:

```
#define Center &fsm[0]
#define Left   &fsm[1]
#define Right  &fsm[2]
#define Left2   &fsm[3]
#define Right2  &fsm[4]

State_t fsm[5]={
  {0x03, 500, { Right, Left,   Right,  Center }},  // Center
  {0x02, 500, { Left,  Left2, Right,  Center }},  // Left
  {0x01, 500, { Right, Left,   Right2, Center }},   // Right
  {0x03, 500, { Left2,  Left, Right2,  Center }},  // Left2
  {0x03, 500, { Right2, Left2,   Right, Center }}   // Right2
```

ITEM	QTY	PART NO.	DESCRIPTION

TITLE: Lab7-2 Finite State Machine

NoSetup KENT, WA 98042 USA	DOCUMENT NUMBER Ti RSLK	REV A
ENGR: NDIEP	SIZE : B	SHEET 1 of 2

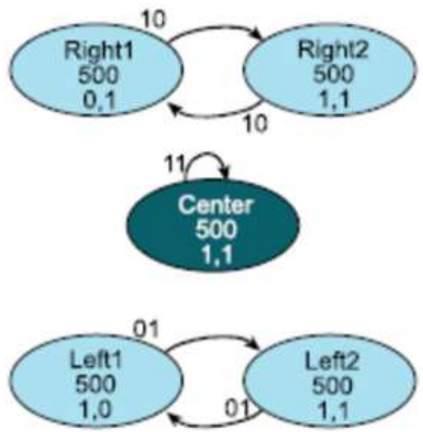


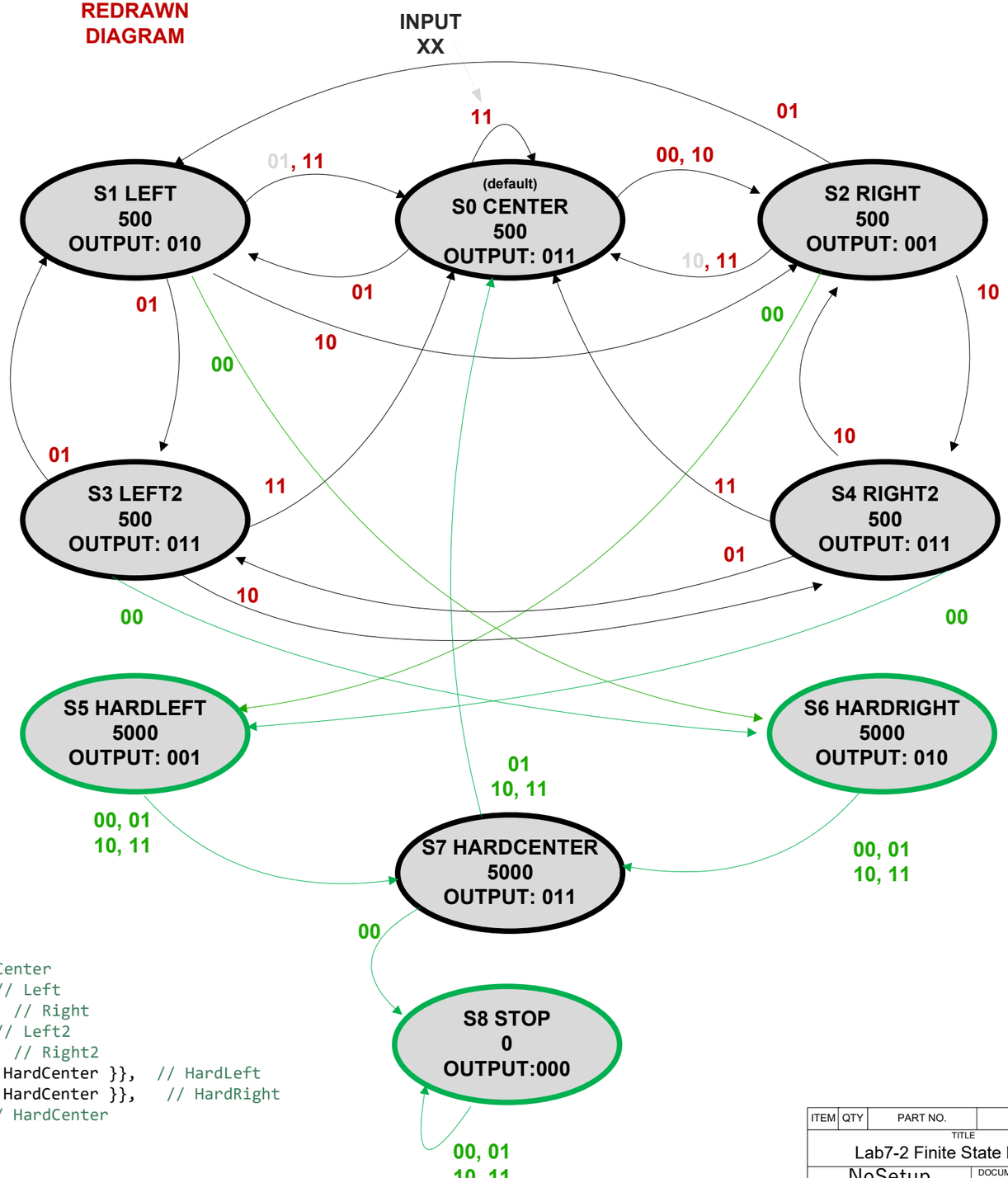
Figure 4. Expanded FSM state graph. The time in each state is shown in 1ms units.

2) The second behavior you need to implement is what happens when the robot goes completely off the line. If it goes off the line to the right (input=0,0 while in Right1 or Right2), it should make a hard left turn (output=0,1) for 5 seconds, then go straight (output=1,1) for 5 seconds. If it is still off the line at this point it should stop (output=0,0). If it finds the line, resume line following. It should take three more states to implement this behavior.

Similarly, if the robot goes completely off the line to the left (input=0,0 while in Left1 or Left2), it should make a hard right turn (output=1,0) for 5 seconds, then go straight (output=1,1) for 5 seconds. If it is still off the line at this point it should stop (output=0,0). If it finds the line, it should resume line following. It should take three more states to implement this behavior.

The solution should have about 11 states (5 states from Figure 4, plus 3 for lost to the right, plus 3 states for lost to the left). As long as you have 9 or more states, feel free to make assumptions or change the exact behavior of the machine. The objective of the lab is to describe the complete behavior of a system with the state transition graph, and then to implement that behavior with a very simple FSM controller. The FSM controller should have NO conditional branch statements.

REDRAWN DIAGRAM



MODIFICATIONS:

```
#define Center &fsm[0]
#define Left   &fsm[1]
#define Right  &fsm[2]
#define Left2   &fsm[3]
#define Right2  &fsm[4]
#define HardLeft   &fsm[5]
#define HardRight  &fsm[6]
#define HardCenter   &fsm[7]
#define Stop  &fsm[8]
```

```
#ifdef LAB7TwoTwo
State_t fsm[9]={
{0x03, 500, { Right, Left,   Right,  Center }},  // Center
{0x02, 500, { HardLeft,  Left2, Right,  Center }},  // Left
{0x01, 500, { HardRight, Left,   Right2, Center }},   // Right
{0x03, 500, { HardLeft,  Left, Right2,  Center }},  // Left2
{0x03, 500, { HardRight, Left2,   Right, Center }},   // Right2
{0x01, 5000, { HardCenter,  HardCenter, HardCenter,  HardCenter }},  // HardLeft
{0x02, 5000, { HardCenter, HardCenter,   HardCenter, HardCenter }},   // HardRight
{0x03, 5000, { Stop,  Center, Center,  Center }},  // HardCenter
{0x00, 000, { Stop, Stop,   Stop, Stop }}   // Stop
};
#endif
```

ITEM	QTY	PART NO.	DESCRIPTION
TITLE: Lab7-2 Finite State Machine			
NoSetup KENT, WA 98042 USA		DOCUMENT NUMBER: Ti RSLK	REV: A
ENGR: NDIEP		SIZE: B	SHEET 2 of 2