Final Presentation

Cell Phone Controlled Security System

16 April 2008

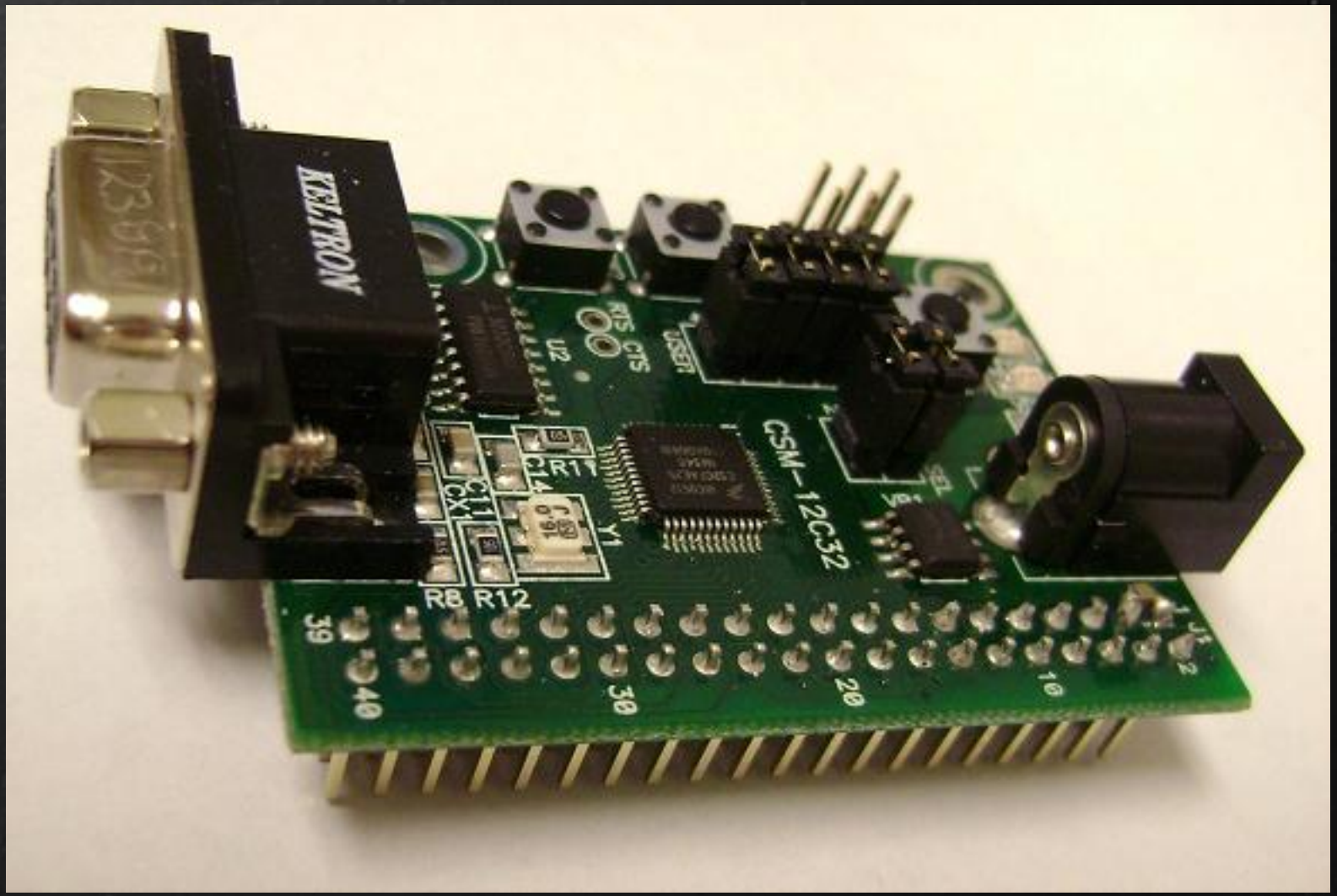
Team Members

Kevin Brown

Don DeLaMare

Brian Faires

6812 Microcontroller



Size 2.2" X 1.6"

6812 Microcontroller

32K Bytes Flash EEProm
2K Bytes Ram
31 I/O Lines
8-Ch 16-bit Timers
SCI/SPI Ports
Key Wake-up port
8MHz Internal Bus
25MHz Operation
40 pin connector
RS-232 Serical Port
3 push buttons (2 user / reset)
3 LEDs (2 user / VDD)

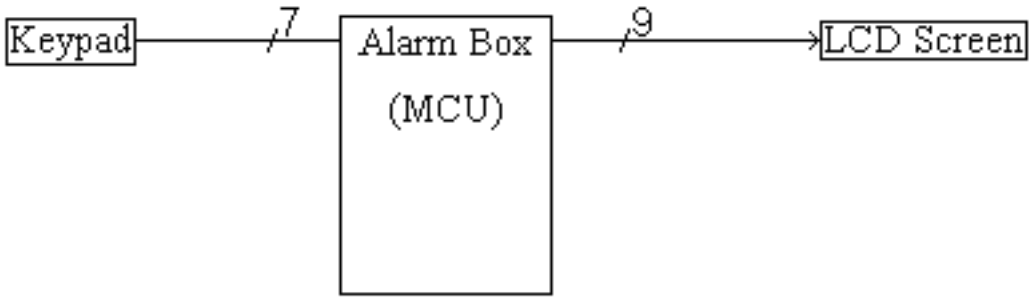
MCU_PORT Connector

	V_x 1	2 PE1/IRQ*	
	GND 3	4 RESET*	
SCI or I/O	PS1/TXD 5	6 MODC/BKGD	
SCI or I/O	PS0/RXD 7	8 NC	
	PP5/KWP5 9	10 NC	
	PE0/XIRQ* 11	12 NC	
	PT0/PW0/IOC0 13	14 NC	
	PT1/PW1/IOC1 15	16 NC	
SPI or I/O	PM4/MOSI 17	18 PAD00/AN00	
SPI or I/O	PM2/MISO 19	20 PAD01/AN01	
SPI or I/O	PM5/SCK 21	22 PB4	
SPI or I/O	PM3/SS* 23	24 PA0	
	PE4/ELCK 25	26 PM1/TXCAN	
	PE7/XCLKS 27	28 PM0/RXCAN	
ADC or I/O	PAD02/AN02 29	30 PT2/PW2/IOC2	General I/O
ADC or I/O	PAD03/AN03 31	32 PT3/PW3/IOC3	General I/O
ADC or I/O	PAD04/AN04 33	34 PT4/PW04/IOC4	General I/O
ADC or I/O	PAD05/AN05 35	36 PT5/IOC5	General I/O
ADC or I/O	PAD06/AN06 37	38 PT6/IOC6	General I/O
ADC or I/O	PAD07/AN07 39	40 PT7/IOC7	General I/O

We know how to use these 24 pins as I/O. We can use 5 more pins.

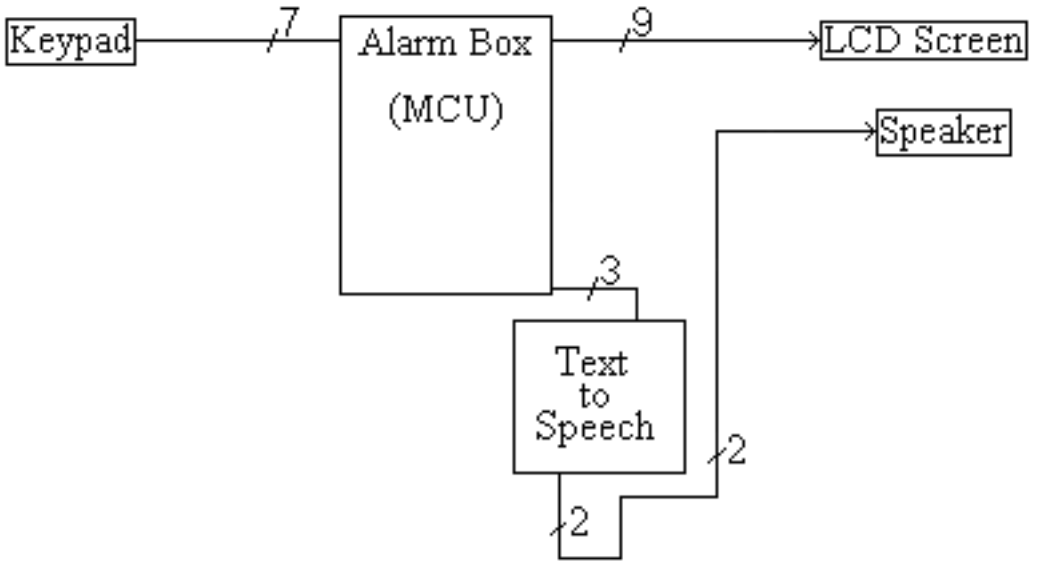
Step 1

Security System



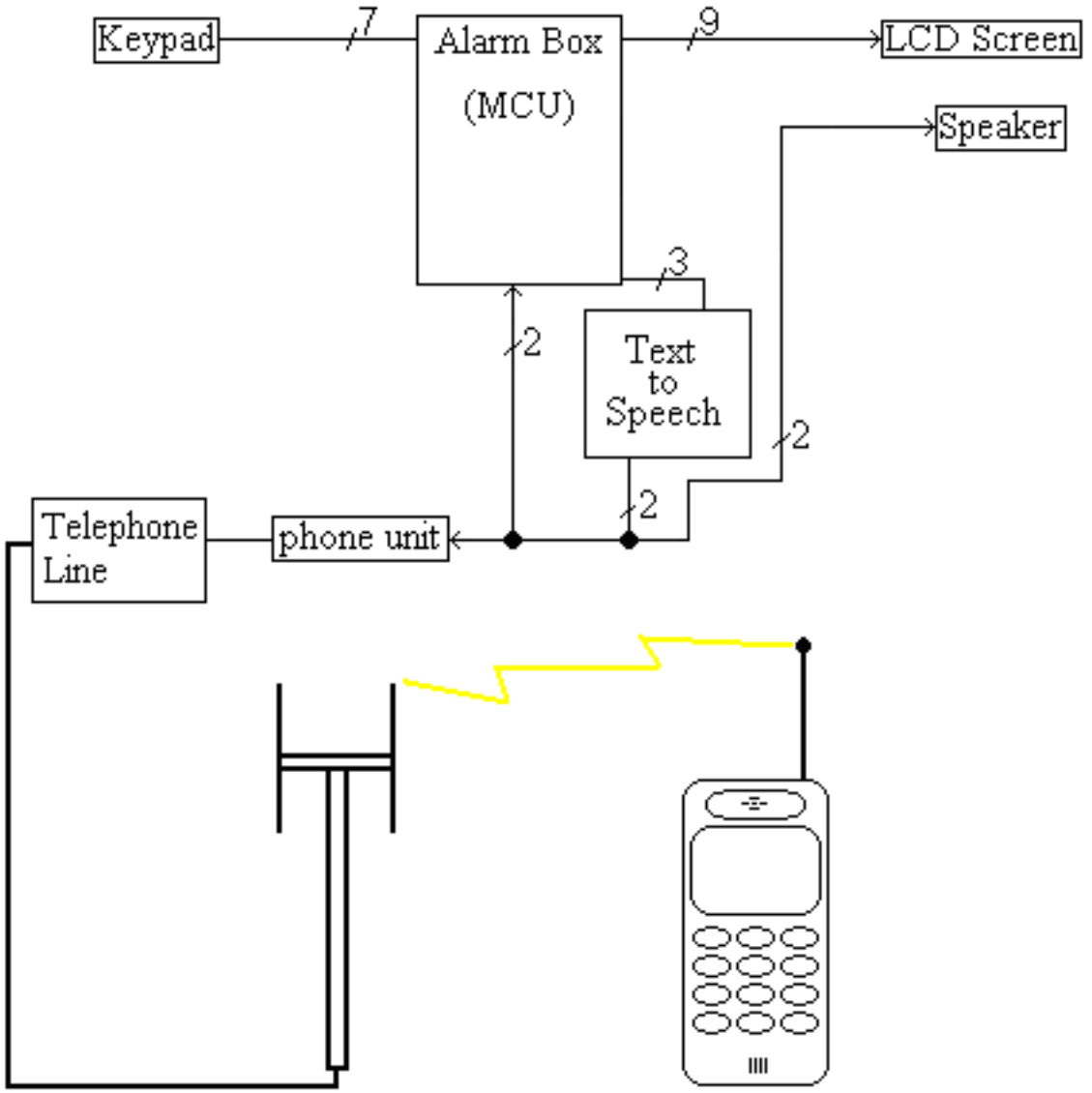
Step 2

Security System



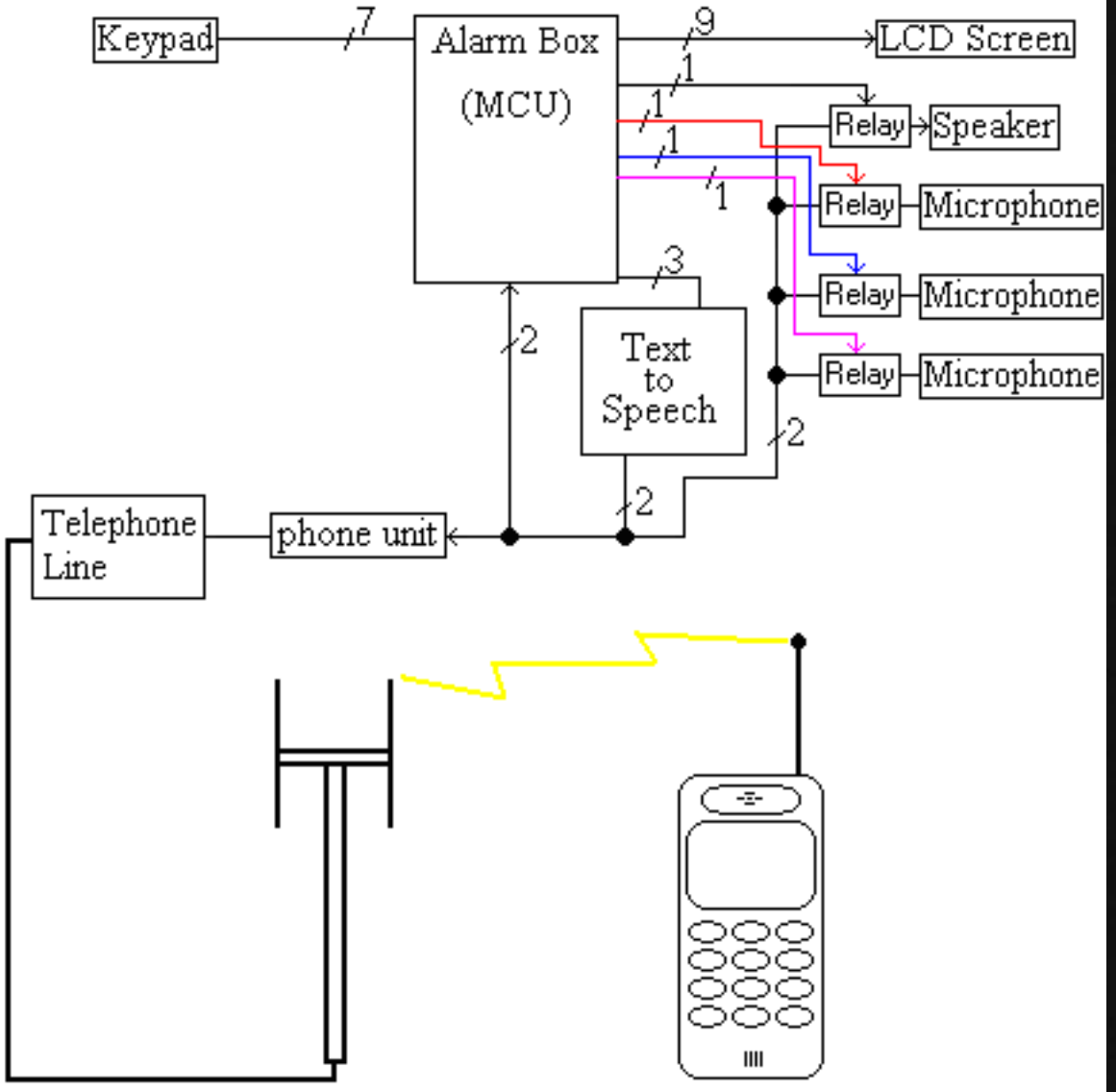
Step 3

Security System



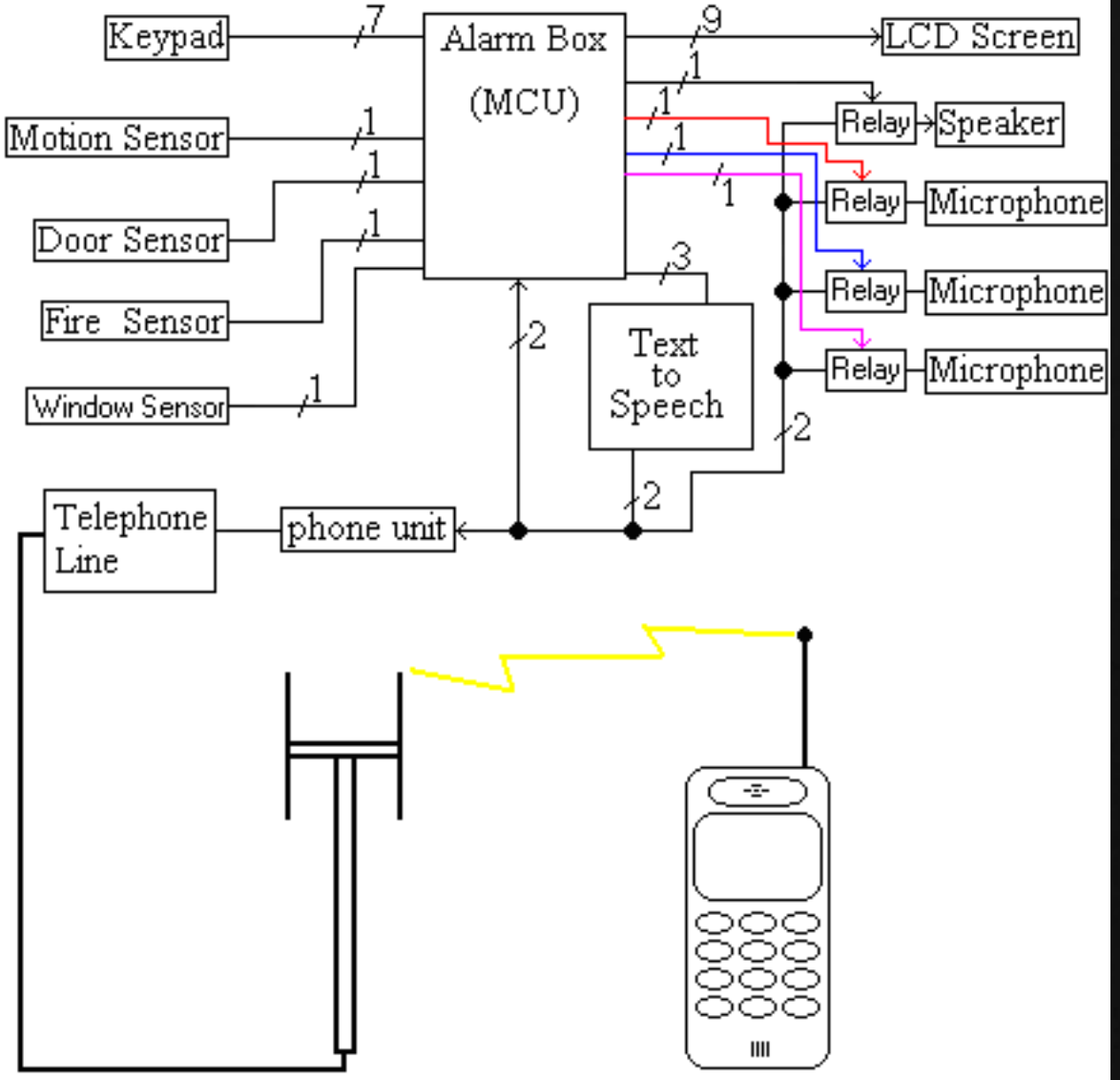
Step 4

Security System



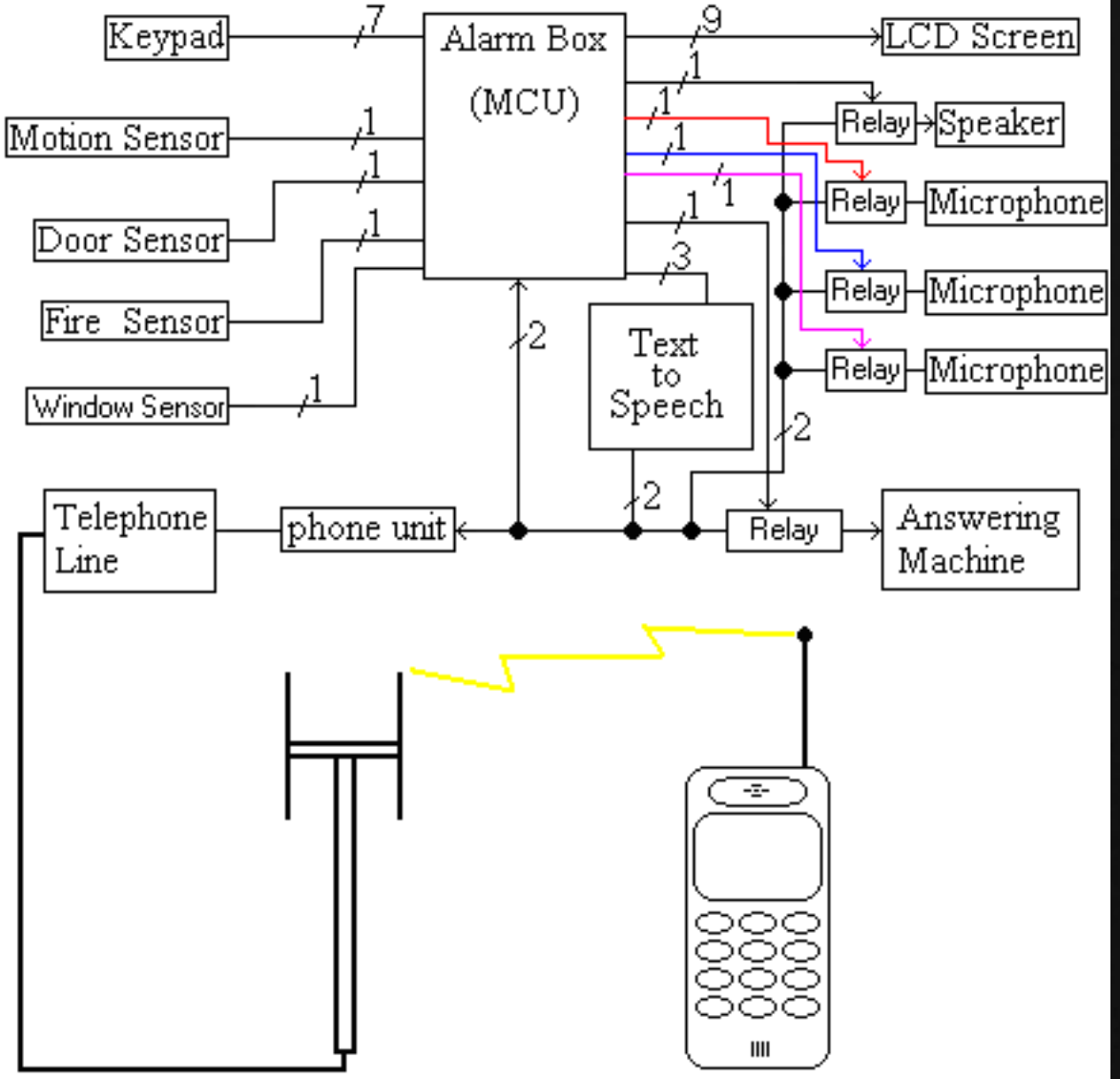
Step 5

Security System

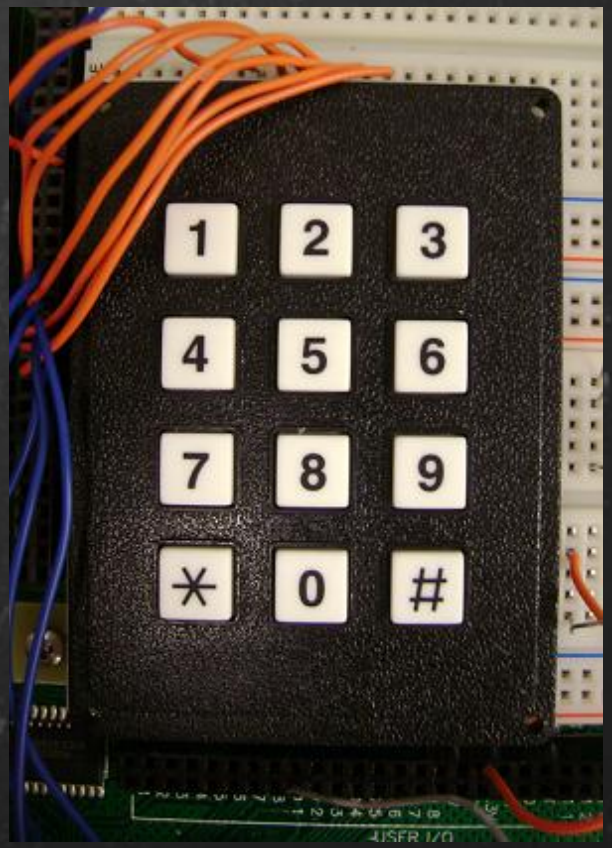


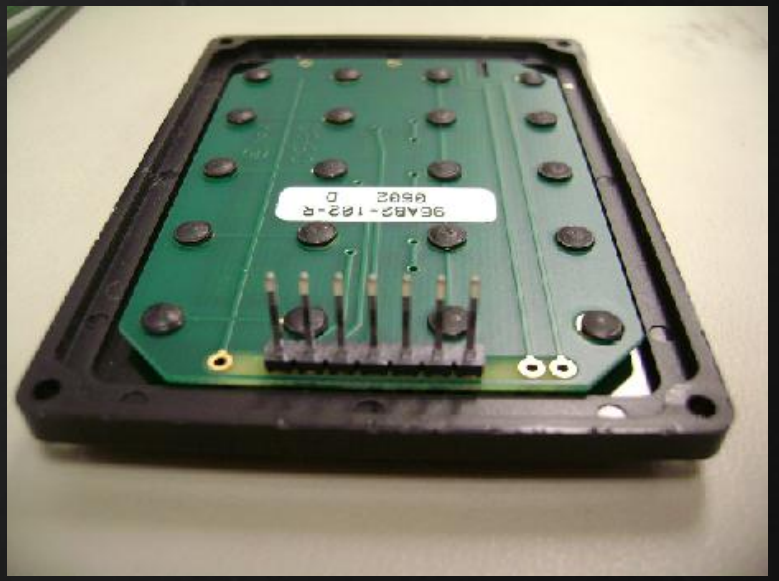
Step 6

Security System



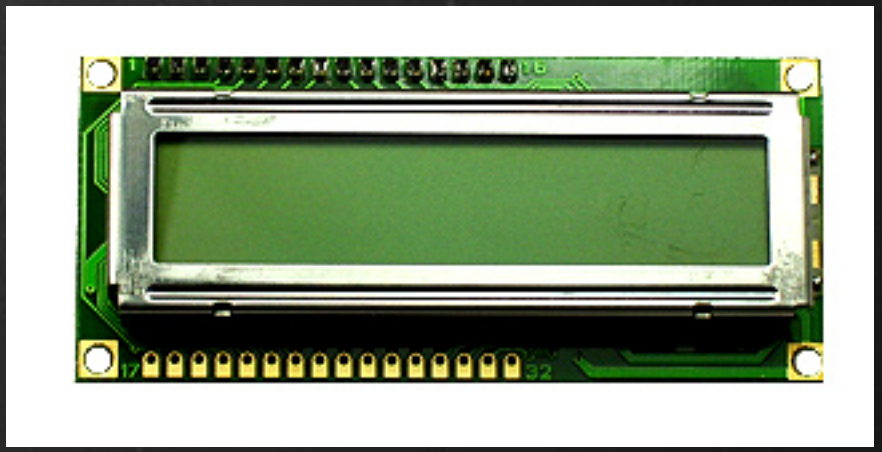
Keypad





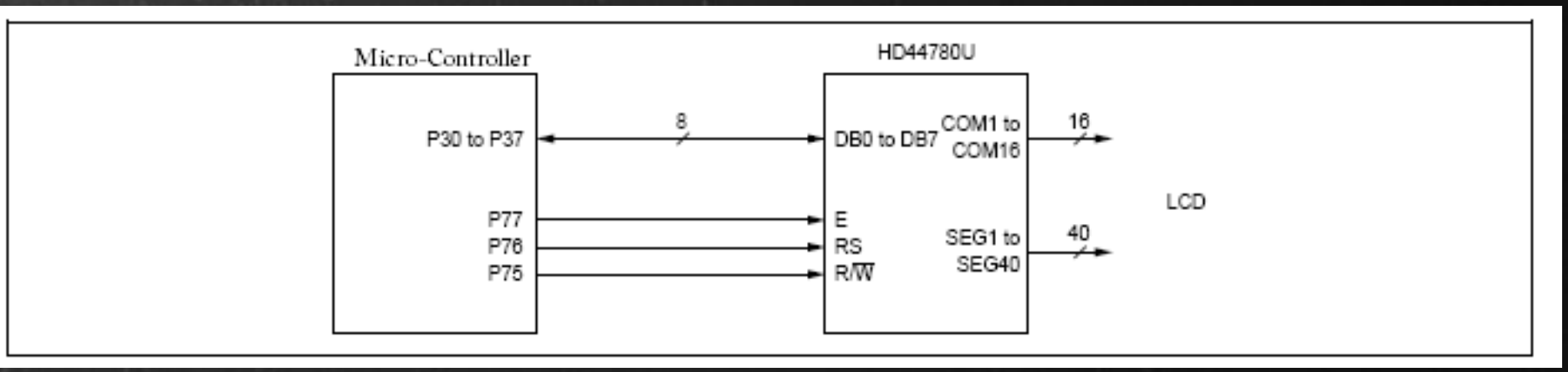
7 parallel wires with four inputs and three outputs. Each key corresponds to a row and a column.

LCD Display - LCD1031 16X2 Characters

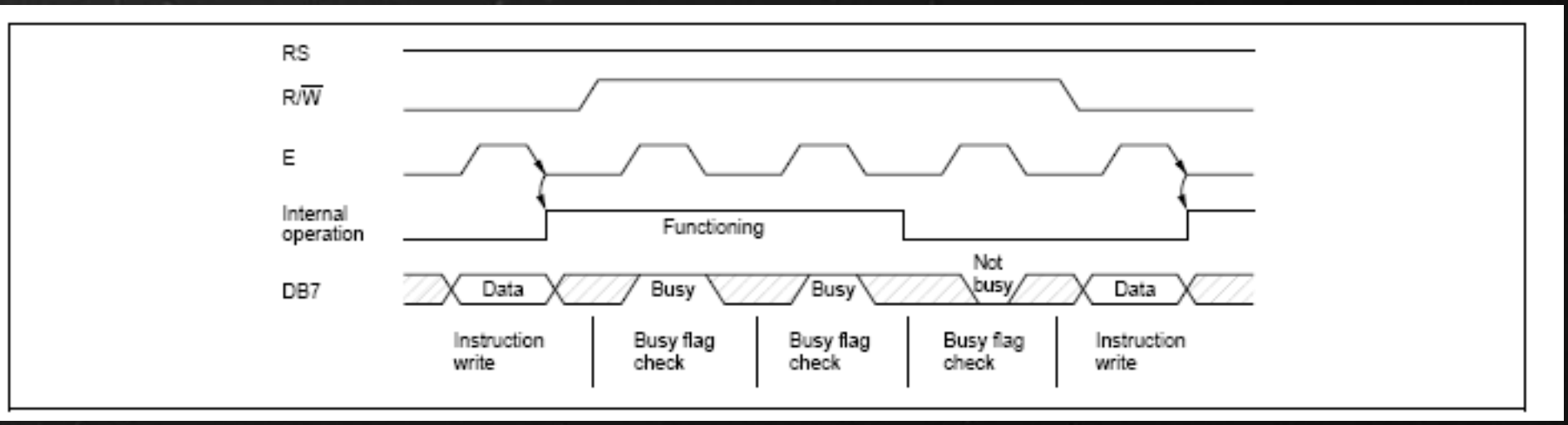


- 16x2 LCD
- Yellow-green LED back-light
- Includes the HD44780 controller
- Machine pin male headers come out of back of board

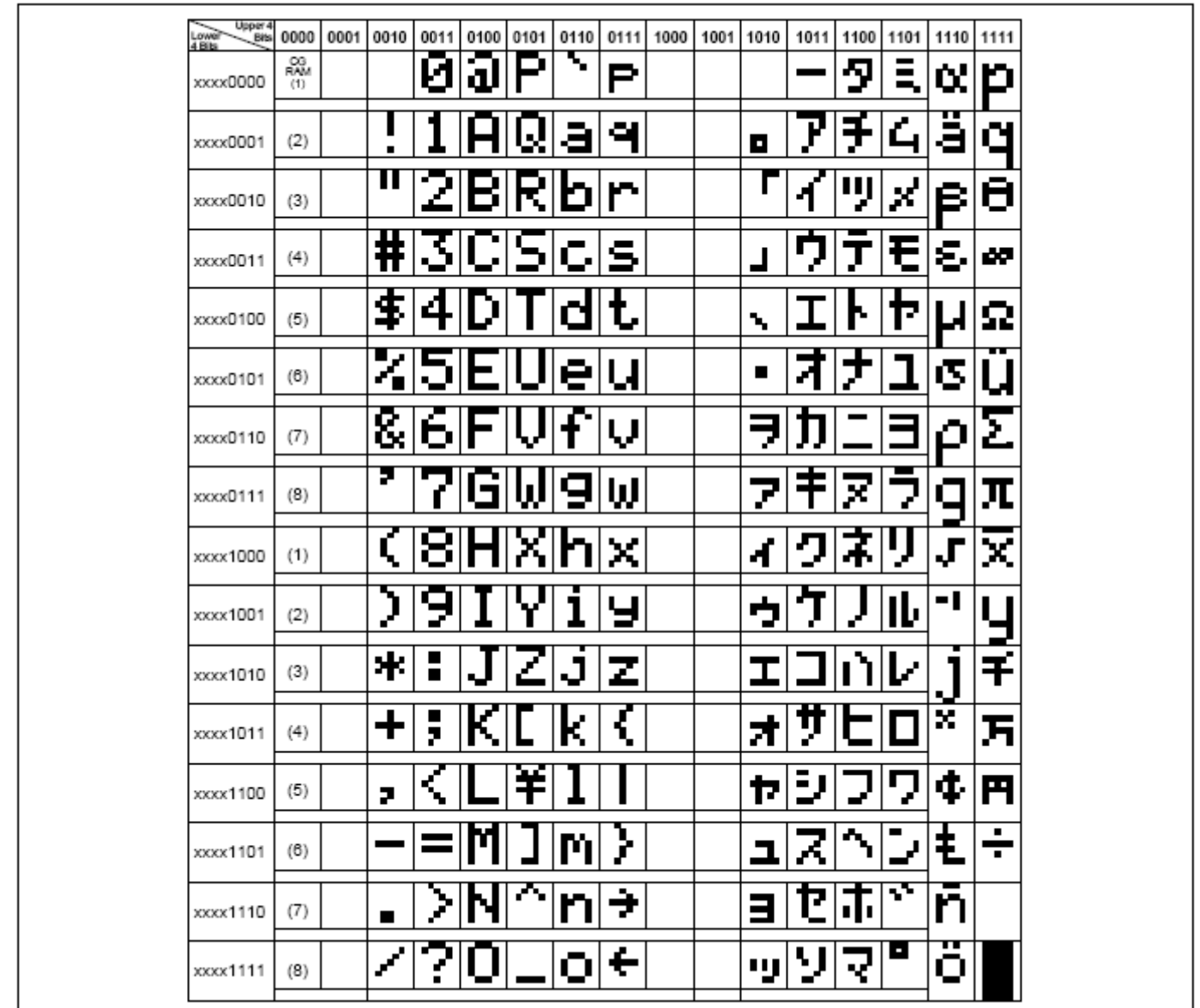
8-bit MCU Interface with LCD Controller



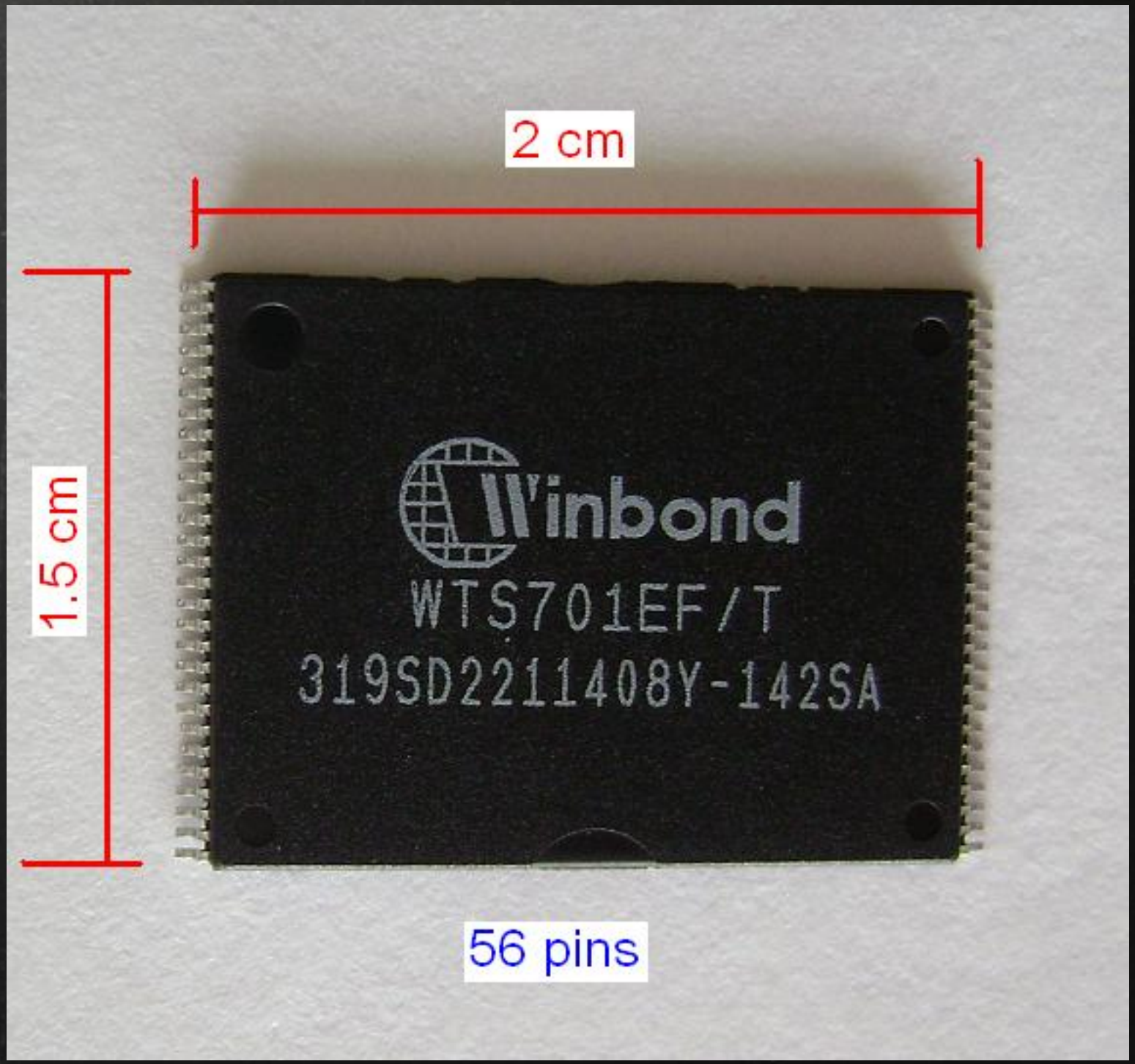
Busy Flag Timing Sequence



HD44780 Character Map



Lower 4 Bits \ Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	@	P	`	p				ー	タ	ミ	α	p
xxxx0001	(2)		!	1	A	Q	a	q			。	ア	チ	ム	ä	q
xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	メ	ß	θ
xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
xxxx0100	(5)		$	4	D	T	d	t			、	エ	ト	ヤ	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	ユ	σ	ü
xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w			ァ	キ	ヌ	ラ	g	π
xxxx1000	(1)		(	8	H	X	h	x			ィ	ク	ネ	リ	√	x̄
xxxx1001	(2)		)	9	I	Y	i	y			ゥ	ケ	ノ	ル	⁻¹	y
xxxx1010	(3)		*	:	J	Z	j	z			ェ	コ	ハ	レ	j	千
xxxx1011	(4)		+	;	K	[	k	{			ォ	サ	ヒ	ロ	ˣ	万
xxxx1100	(5)		,	<	L	¥	l	|			ャ	シ	フ	ワ	¢	円
xxxx1101	(6)		−	=	M	]	m	}			ュ	ス	ヘ	ン	ŧ	÷
xxxx1110	(7)		.	>	N	^	n	→			ョ	セ	ホ	゛	ñ	
xxxx1111	(8)		/	?	O	_	o	←			ッ	ソ	マ	°	ö	


2 cm
1.5 cm
Winbond
WTS701EF/T
319SD2211408Y-142SA
56 pins

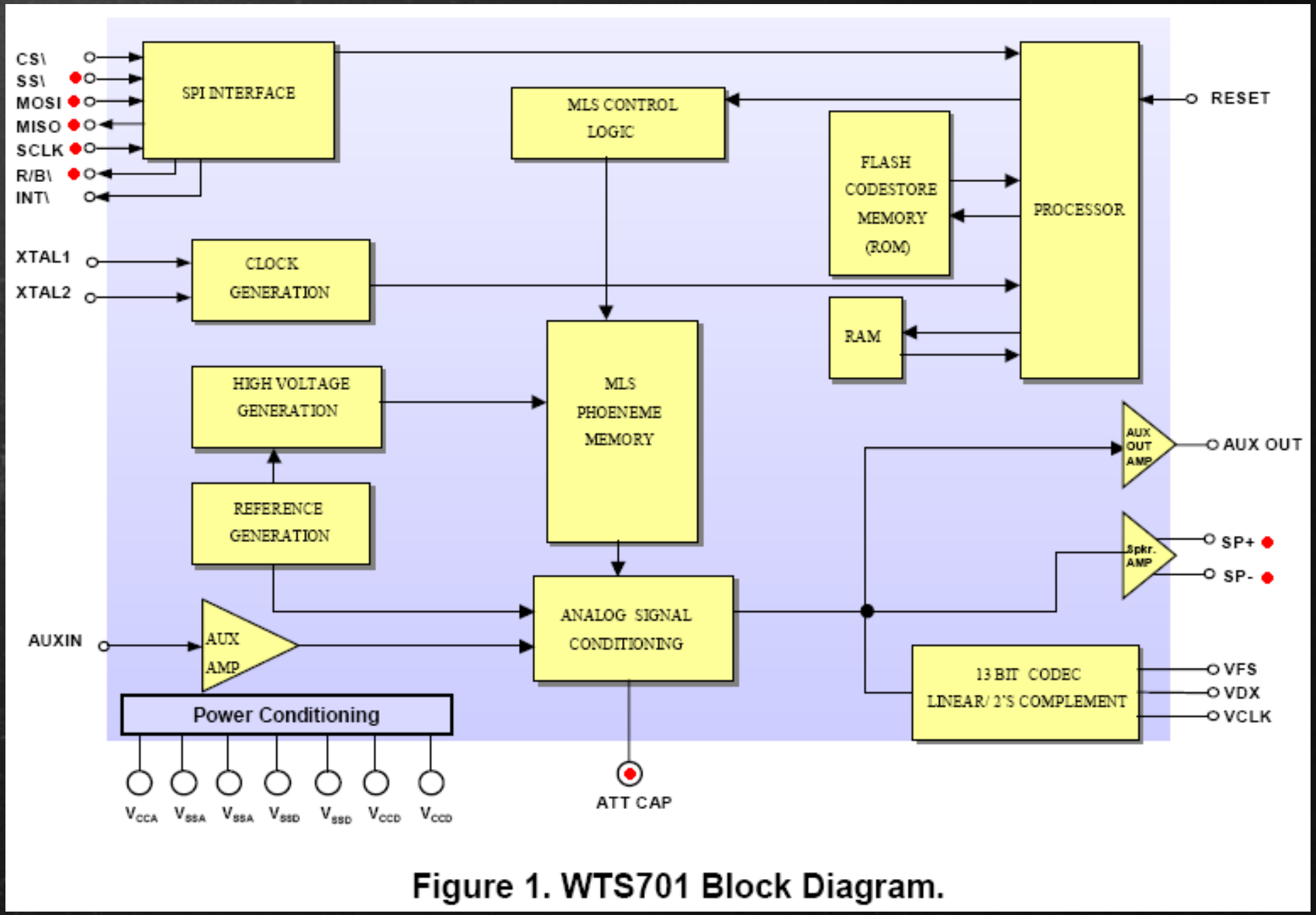


Figure 1. WTS701 Block Diagram.

MCU interface with Text to Speech chip

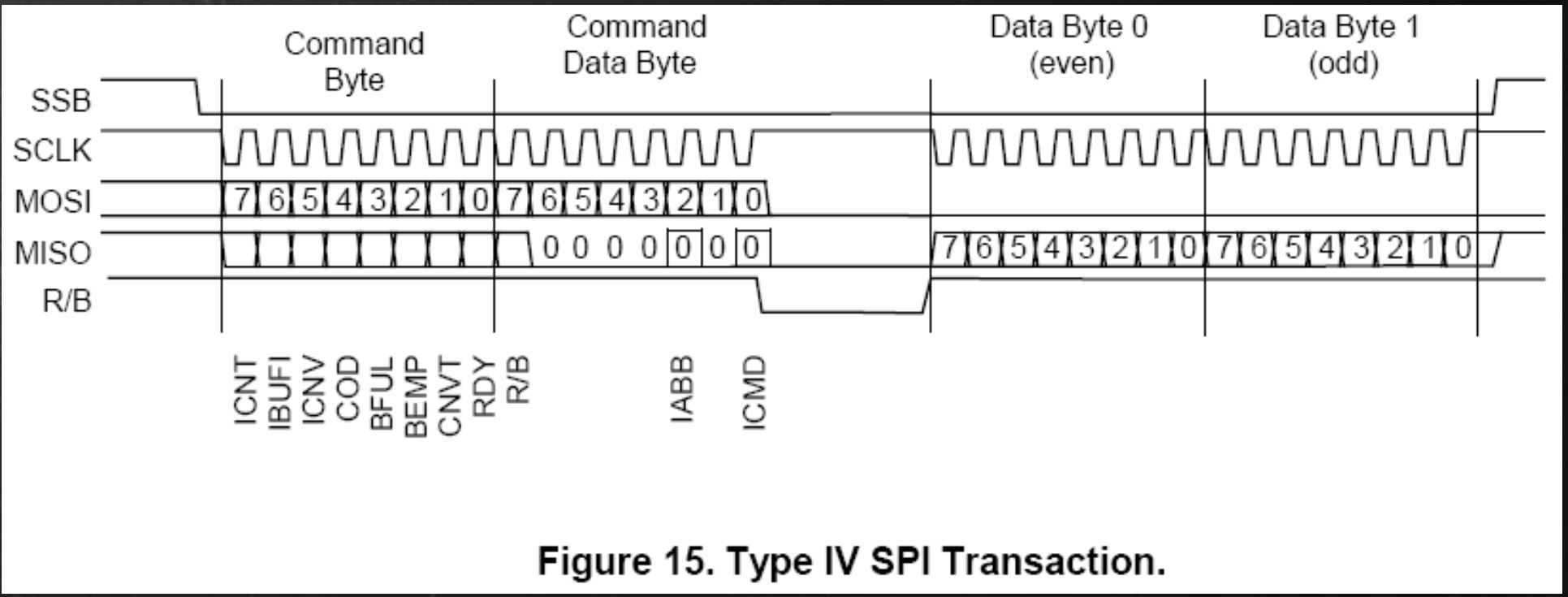


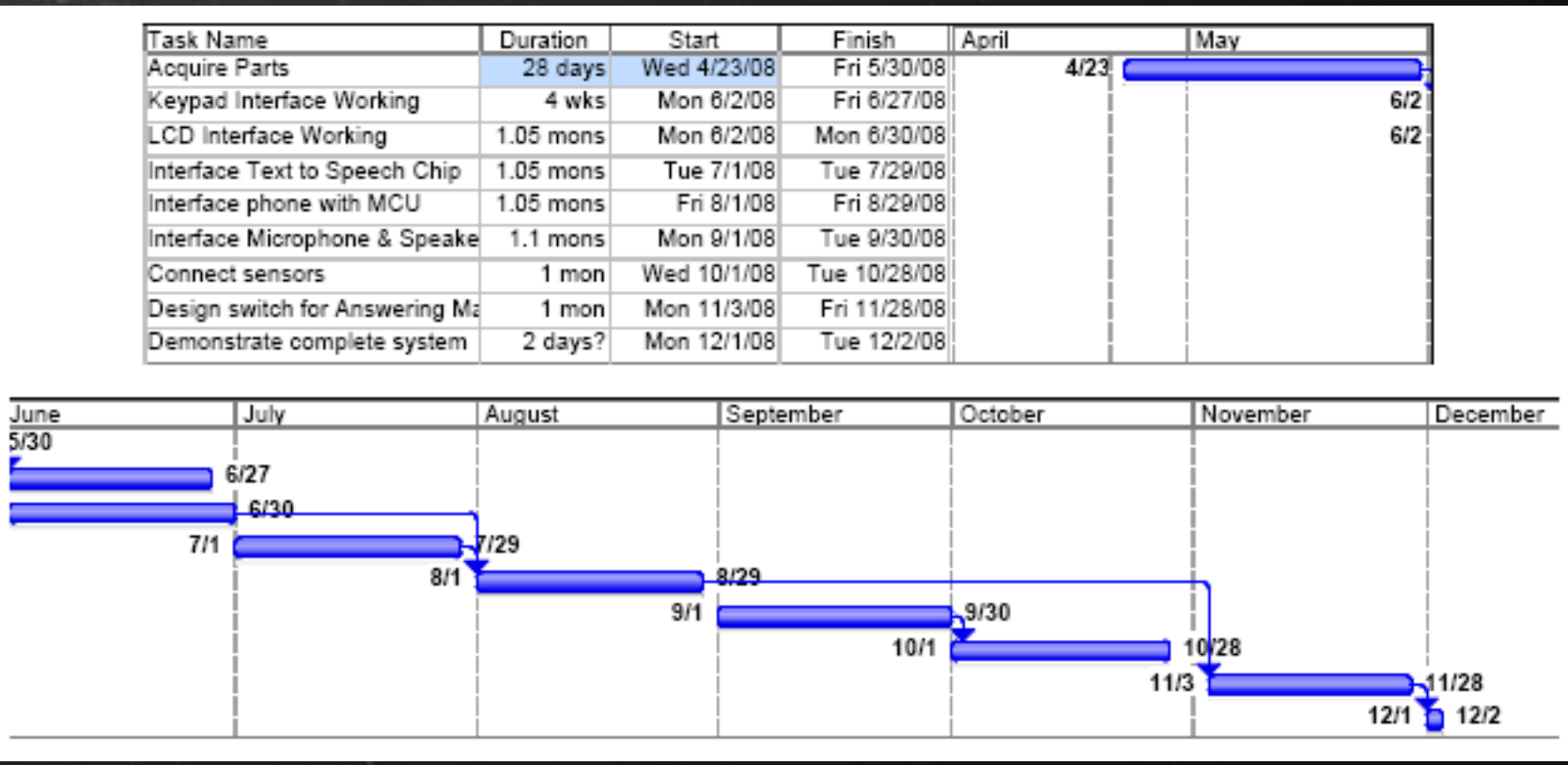
Figure 15. Type IV SPI Transaction.

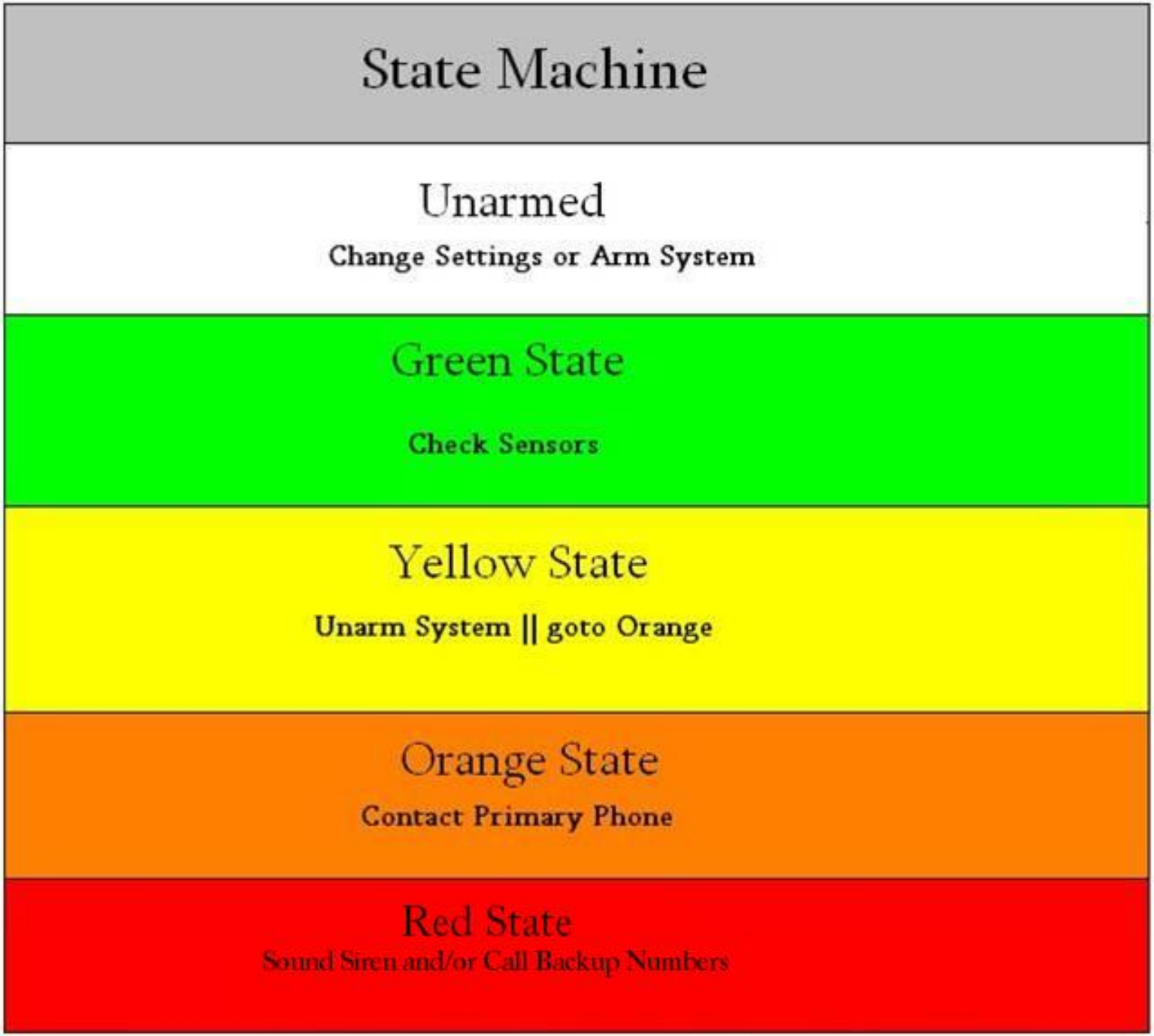
MCU interface with Phone Line

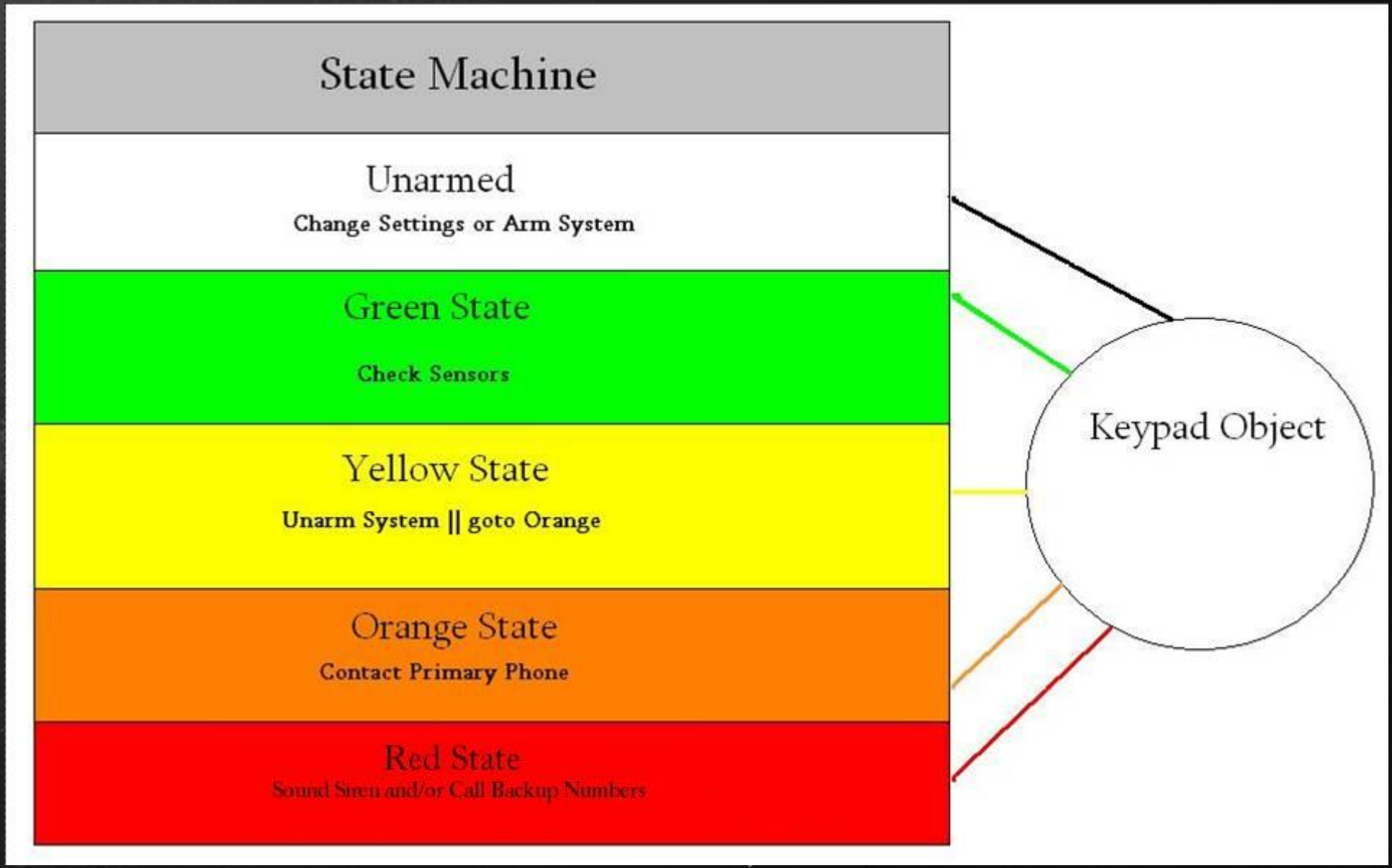
The microcontroller will have to pick up the phone on incoming calls if there is no answering machine. The microcontroller will constantly listen to the phone line for a ring (high voltage pulse) and answer the phone (allowing a current to flow back on the phone line) after the desired number of rings. Then it will send out a string to the text to speech chip to say, "enter code." Then the microcontroller will listen for a low and a high frequency to detect a row and a column of a four by four matrix to match a button press. If the correct code was entered, it will go to the menus of the security system, then continue sending the menus' strings to the text to speech chip and listen for button presses to switch between menus.

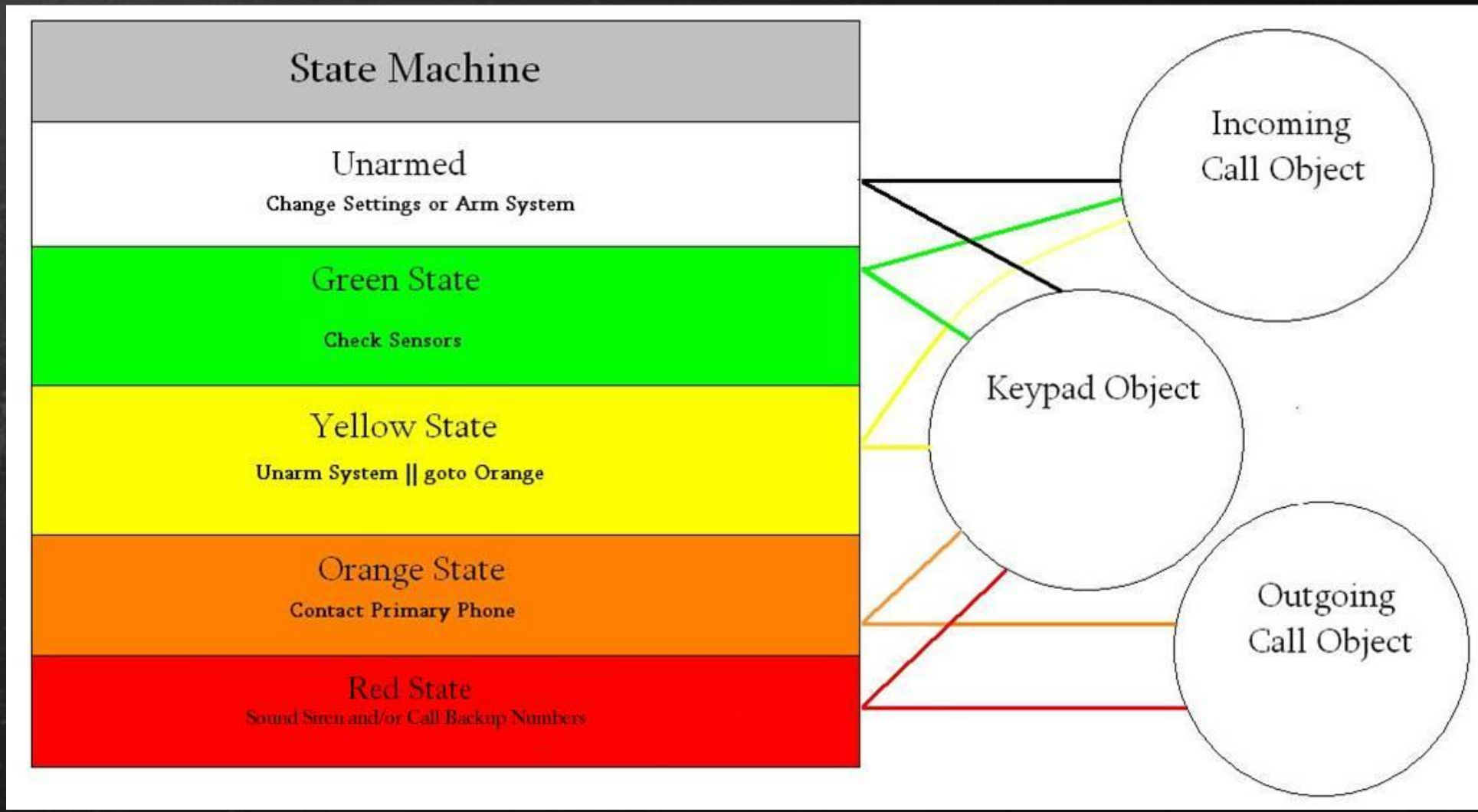
Schedule

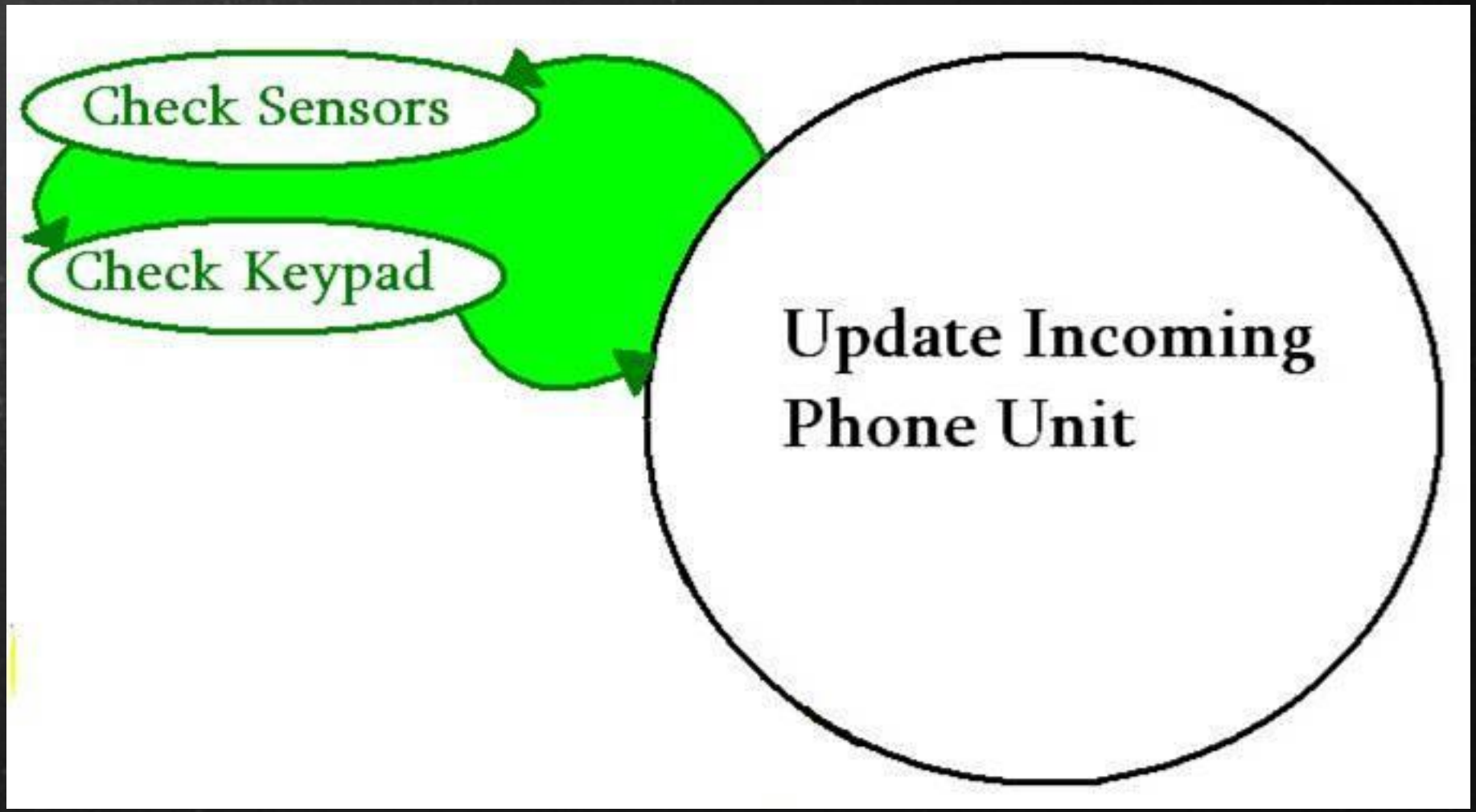
Task Name	Duration	Start	Finish
Acquire Parts	28 days	Wed 4/23/08	Fri 5/30/08
Keypad Interface Working	4 wks	Mon 6/2/08	Fri 6/27/08
LCD Interface Working	1.05 mons	Mon 6/2/08	Mon 6/30/08
Interface Text to Speech Chip	1.05 mons	Tue 7/1/08	Tue 7/29/08
Interface phone with MCU	1.05 mons	Fri 8/1/08	Fri 8/29/08
Interface Microphone & Speake	1.1 mons	Mon 9/1/08	Tue 9/30/08
Connect sensors	1 mon	Wed 10/1/08	Tue 10/28/08
Design switch for Answering Ma	1 mon	Mon 11/3/08	Fri 11/28/08
Demonstrate complete system	2 days?	Mon 12/1/08	Tue 12/2/08

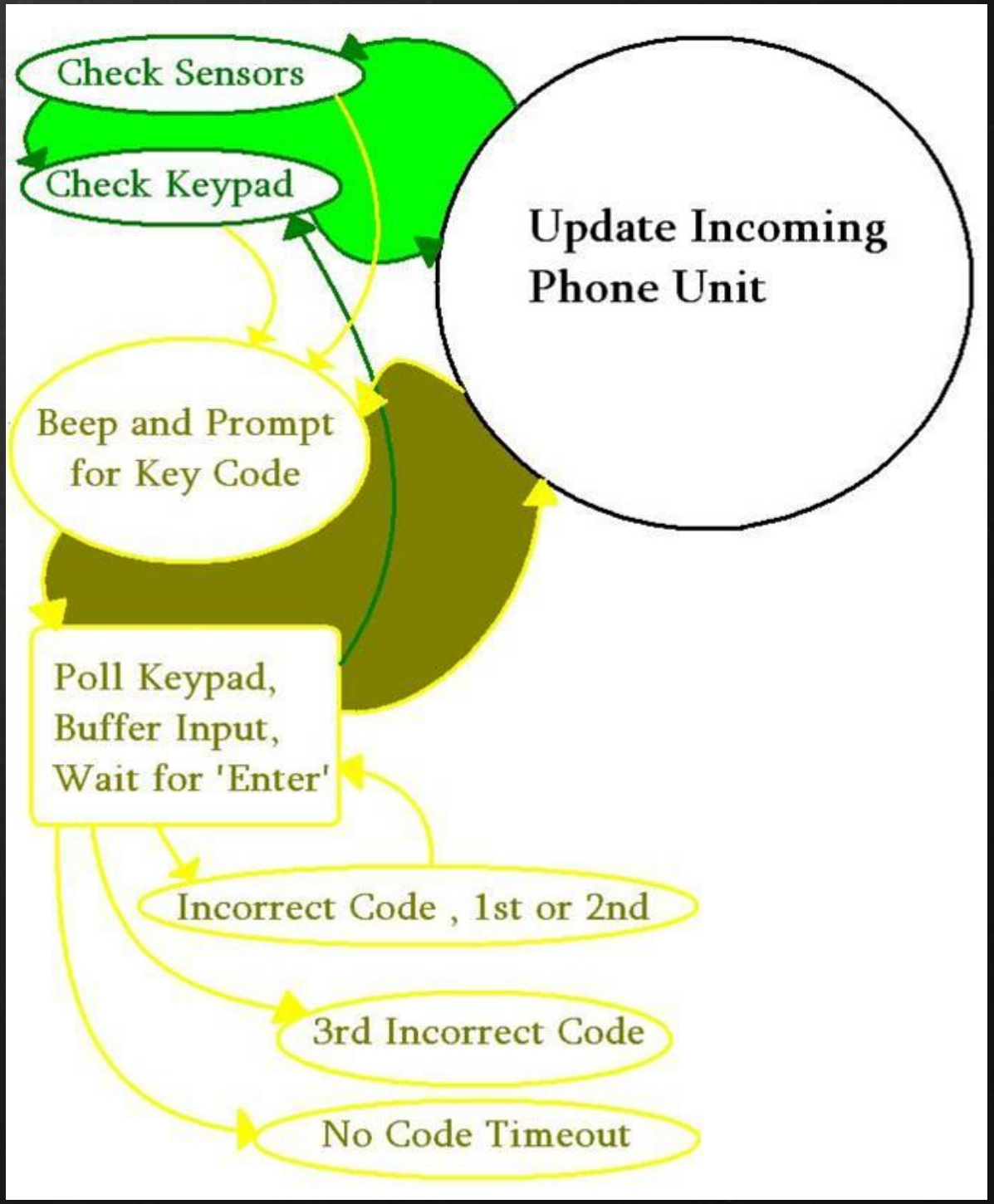


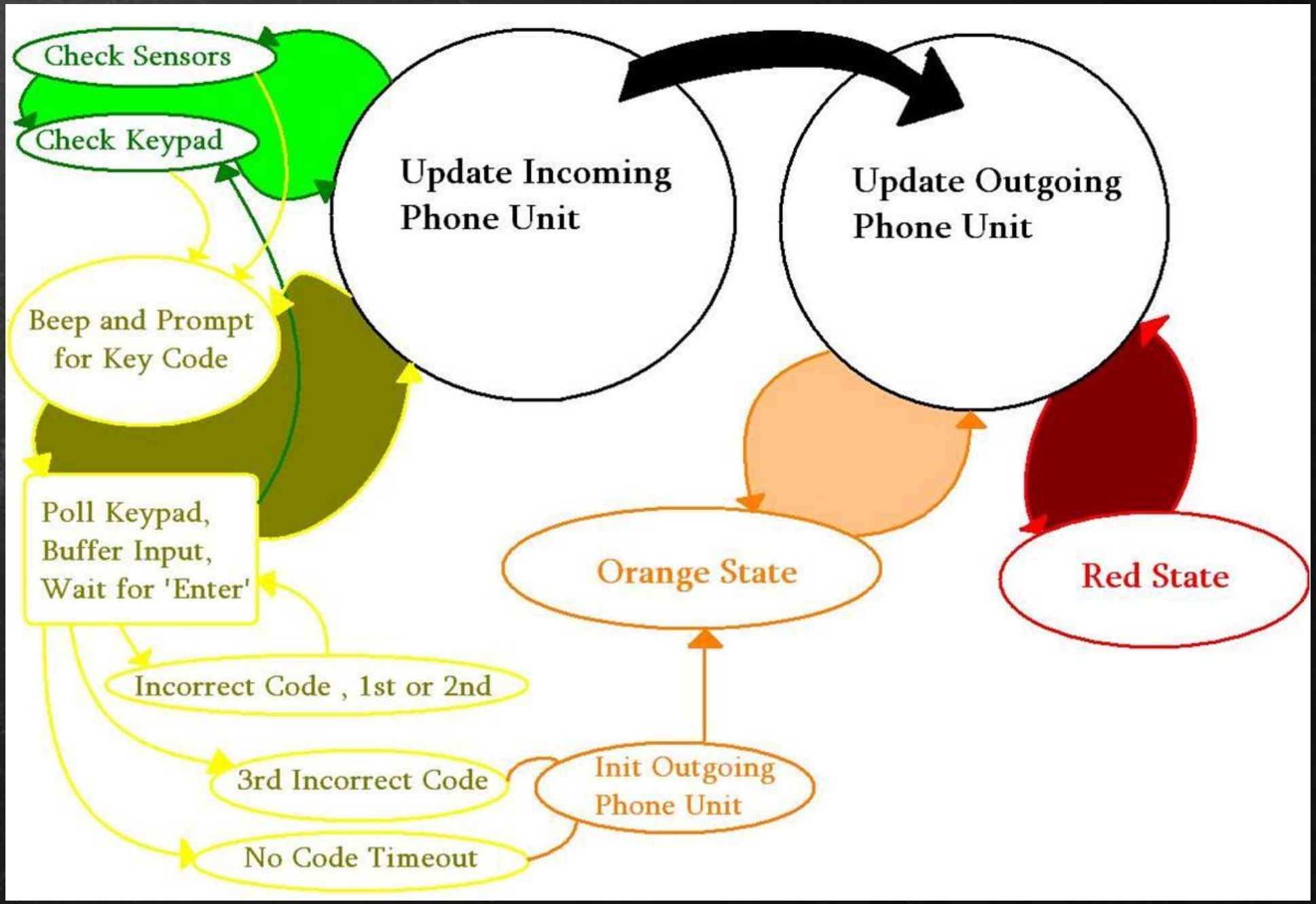

State Machine
Unarmed
Change Settings or Arm System
Green State
Check Sensors
Yellow State
Unarm System || goto Orange
Orange State
Contact Primary Phone
Red State
Sound Siren and/or Call Backup Numbers

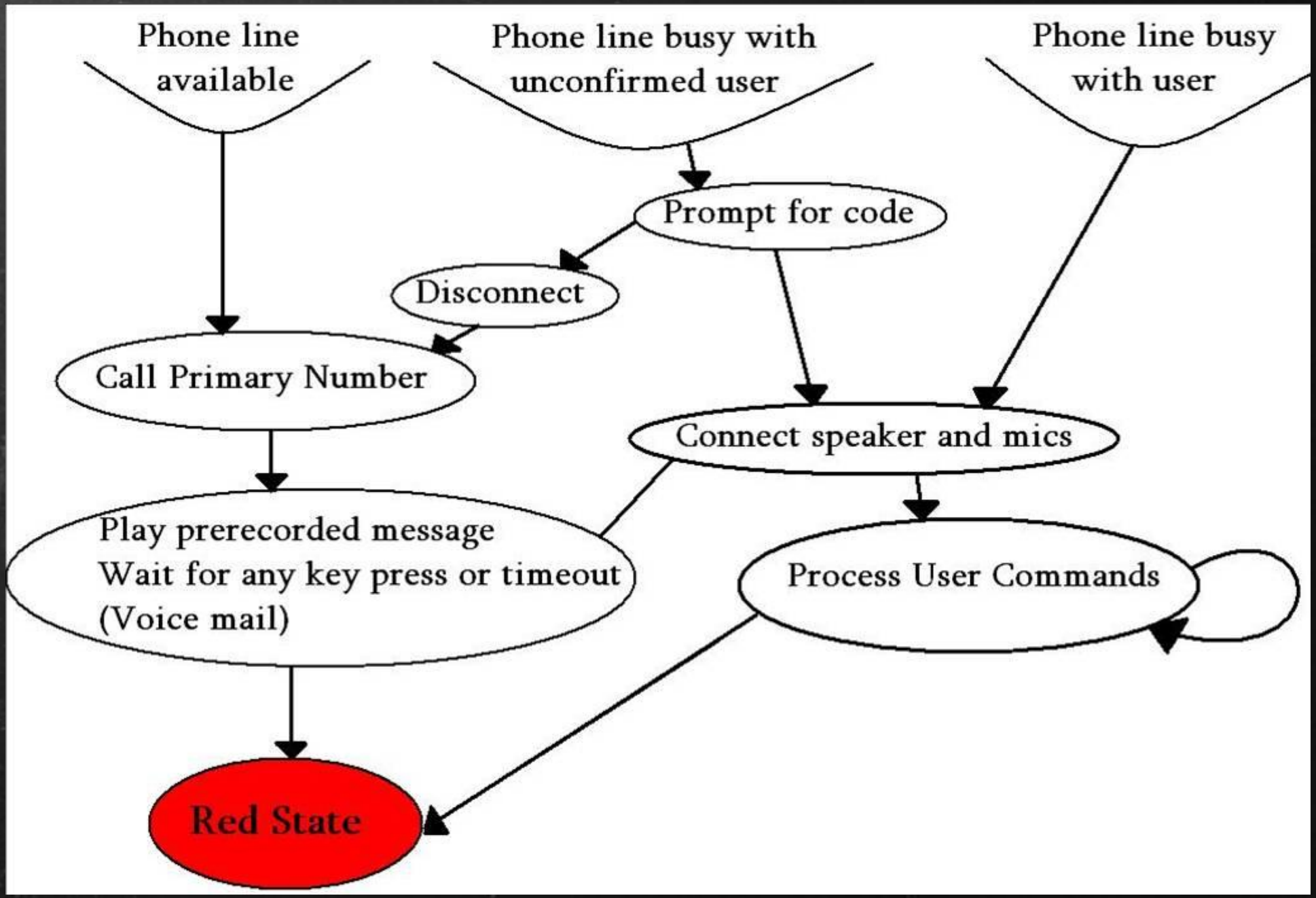

State Machine
Unarmed
Change Settings or Arm System
Green State
Check Sensors
Yellow State
Unarm System || goto Orange
Orange State
Contact Primary Phone
Red State
Sound Siren and/or Call Backup Numbers
Keypad Object


State Machine
Unarmed
Change Settings or Arm System
Green State
Check Sensors
Yellow State
Unarm System || goto Orange
Orange State
Contact Primary Phone
Red State
Sound Siren and/or Call Backup Numbers
Incoming Call Object
Keypad Object
Outgoing Call Object

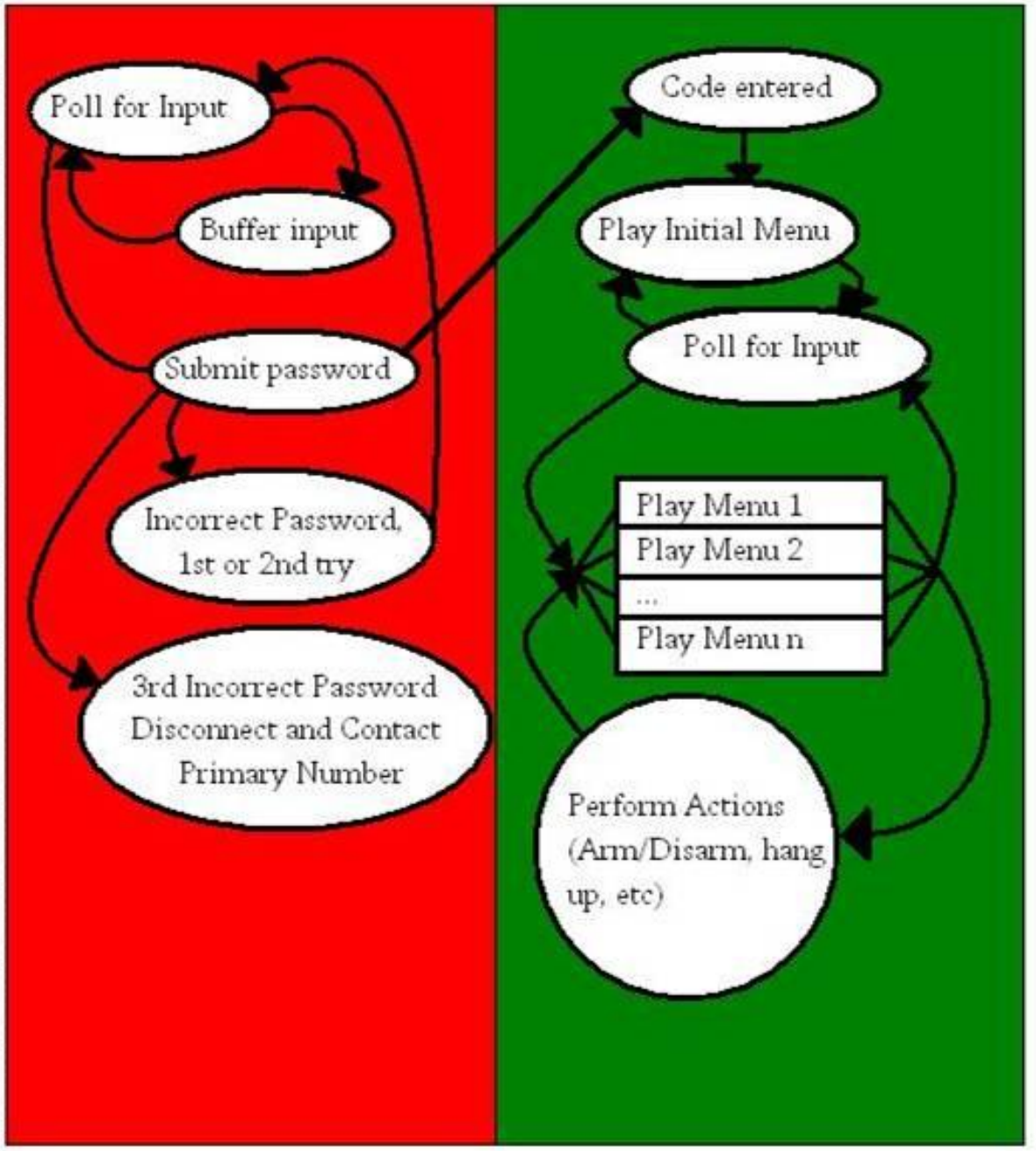

Check Sensors
Check Keypad
Update Incoming
Phone Unit


Check Sensors
Check Keypad
Update Incoming Phone Unit
Beep and Prompt for Key Code
Poll Keypad, Buffer Input, Wait for 'Enter'
Incorrect Code , 1st or 2nd
3rd Incorrect Code
No Code Timeout


Check Sensors
Check Keypad
Update Incoming Phone Unit
Update Outgoing Phone Unit
Beep and Prompt for Key Code
Poll Keypad, Buffer Input, Wait for 'Enter'
Orange State
Red State
Incorrect Code , 1st or 2nd
3rd Incorrect Code
Init Outgoing Phone Unit
No Code Timeout


Phone line available
Phone line busy with unconfirmed user
Phone line busy with user
Prompt for code
Disconnect
Call Primary Number
Connect speaker and mics
Play prerecorded message
Wait for any key press or timeout
(Voice mail)
Process User Commands
Red State

Incoming Call Unit



Risks

Phone line input

- How quickly will our MCU be able to detect a button press?
- Will quickly pressing one key multiple times be detected as a single keypress?

LCD Support chip

- Needed for 16x2. We won't know if we can do this until we have the LCD display. May have to stick with 8x2 or 16x1.

Risks cont.

- Shortage of available pins: We will aim to use serial/serial to parallel communication where possible. If there is still a shortage we can use encoders and decoders.

- Program size: If the program cannot fit on 32KB of EEPROM, we may have to replace some of the user specified options with predefined functionality.

Risks cont.

- We want the user to be able to cut off a phone menu if they please. Since we don't know of a way to tell the TTS chip to 'stop talking' we may have to send it words at the rate of speech or ignore button presses mid-sentance.

Security System