

File **DishwasherV1-8.BS1**
Purpose **Replacement Dishwasher brain**
Author Charles Brown
E-mail Dsumdawgy@yahoo.com
Started 27 Jan 2010
Beta Ends 21 Apr 2010
Updated 28 Apr 2010
Comment Do NOT attempt if you haven't had any HI-Voltage-AC safety training.



Disclaimer: I do NOT take any responsibility towards ANY use of this project. This is because, I have no control over your situation, ability or expertise. As a direct result.... **Any application you make of any part of this project is, AT YOUR OWN RISK.**

<<<<<Thanks to Nuts & Voltz!! & my Brother Mitch>>>>>

Table Of Contents

<u>Page</u>	<u>Contents</u>
001	Cover & Table of Contents
002	Intro & v1.7 EEPROM map
003	Operation Manual
004	Stamp Board Info & Schematic
005	Shift Board data
006	Shift Board Schematic
007	Front Panel info
008	Relay Harness info
009	Images of Brain Housing
010	BS1-IC relevant data
011	IC1 (Serial Shifter) relevant data
012	IC2 (Darlington array) relevant data
012	DishwasherV1-7.BS1 (program listing)
TBA	Dishwasher Shell relevant data

Intro

The challenge.....

To build a working feature rich "Dishwasher Brain". Using & constrained by, a BS1-IC. As it turns out, a "fairly" feature rich unit is difficult but not impossible. From the start it was obvious that at least one, 8 bit Parallel to serial IC was required. While that solves the output pins dilemma, the program space was another issue. Avoiding IF-Then usage by using branch & lookup instead, helped. 'Resetting output data each prog loop also made tighter code. With each Loop, the bits are reset to zero. Each run then only has to set the bits. This is not normally the method used in a program. But, to save space, this method turned out to be required. In the end, the Eeprom is fairly well packed. (100% packed see below. Or load the program into your own editor.). However, I did manage to give the unit 6 modes.

- 1---Drain (03.0 minutes)
 - 2---Rinse Only (15.0 minutes)
 - 3---Wash (46.5 minutes)
 - 4---HeavyWash (63.0 minutes)
- (5&6 repeat 3&4 respectively.. But with a 1.5 hour start delay)

I only need this device for one unit. So, I don't intend to re-work/rebuild the boards/housing. I consider this application, a working prototype.

As of 05/19/10, I have these remaining issues:

- 1) Door sensor RJ-11 wire limits current, pull-up resistor needs an increased value. Reads open randomly w/closed..Reads open correctly w/open.
- 2)BS1 self reset not yet tested/functional.
- 3)Random unpredictable RFI resets causing improper program termination.
- 4) Working on V2.0 of code. Lookup of word data. (Lbyte=settings,Hbyte=time). Allowing for far more precise sequence timing. May have to sacrifice modes 5&6 to finalize fitting code into Eeprom.

What follows...is/are the fruits of my labor/insanity.

Memory Map - EEPROM 99% Full (DishwasherV1-8.bs1)

EEPROM Map

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	00	80	CD	0A	E1	6F	AC	14	69	3D	B1	DB	BC	48	94	56
010	6C	22	01	5C	6A	9B	6A	E9	1E	D9	4F	52	25	CB	AA	A7
020	CC	6A	E7	28	6D	C4	56	05	E2	D8	6E	AC	11	21	5D	89
030	BA	A8	4F	9A	67	39	B3	F2	3C	67	AB	A2	AE	B6	C6	18
040	5B	93	15	2D	56	51	13	29	39	84	58	13	79	B5	E1	8C
050	9A	F5	B0	63	A5	35	66	B5	3E	6B	56	87	1D	68	3B	14
060	69	42	CE	DD	BD	BD	BD	BD	9D	E4	DB	5A	51	65	35	
070	91	85	8E	D5	A4	59	C5	96	95	0B	12	EC	21	1F	32	C1
080	52	9A	0B	12	EC	21	13	2C	A5	9F	98	5D	40	50	29	8E
090	A8	10	11	A3	A6	AE	9E	96	AE	9E	9E	9E	96	8A	2A	1B
0A0	D1	CC	05	22	8D	9A	2A	AA	EC	61	73	95	90	B4	CD	6B
0B0	E9	07	71	72	2B	6A	51	B1	EA	21	B3	8A	32	EB	61	B3
0C0	40	49	F5	B0	A1	B4	62	53	66	75	D8	92	D6	99	12	2A
0D0	68	6A	D0	44	16	3A	56	82	4A	6B	44	83	CA	63	6C	7E
0E0	6F	97	60	75	35	2F	28	6D	E6	F5	08	11	6B	14	92	C2
0F0	DB	BC	A0	0F	B7	A2	36	D5	6A	C1	89	F0	C8	75	FB	01

RAM Map

PORT: [Red bar]

W0: [Green bar]

W1: [Green bar]

W2: [Green bar]

W3: [Green bar]

W4: [Green bar]

W5: [Green bar]

W6: [Yellow bar]

RAM Legend

- [Red] - Pins
- [Blue] - Word
- [Green] - Byte
- [Purple] - Bit
- [Yellow] - GOSUB
- [Grey] - Unused

EEPROM Legend

- [Dark Blue] - Undef. Data
- [Light Blue] - Def. Data
- [Medium Blue] - Program
- [White] - Unused

Source Code: DishwasherV1-8.bs1

Close

Operation Manual for Replacement Dishwasher Brain

Setting Wash mode

Press & hold "Start". Initially, there will be 2 beeps this indicates "mode 2".

Release "Start" during the sounding of the mode you desire.

(Example: During the sounding of 4 beeps release "Start" to select "mode 4")

Once mode count has reached it's maximum, it will return to "mode 1" & continue to count up to maximum. If user accidentally moves past their desired mode, they can simply keep start depressed until the count loops and the desired mode is again sounded. [The loop has no limit and will continue until user releases "Start".]

NOTE: With all program loadouts, Mode1 (1 beep) will always be Off/Abort

In case user accidentally releases "Start" too soon, there is a delay of 3 seconds before a mode is activated. This delay allows user to depress "Start" again to continue through mode selection.

Stopping/Resetting Dishwasher

Press both "Start" & "HI-Temp" simultaneously, the unit will sound a short series of beeps & wait for a mode to be selected.

NOTE: When the unit is reset. It will NOT automatically drain the washwater.

Mode Selection v1.7 &1.8

1---Off	
1---Drain	(03min)
3---Rinse	(15min)
4---Wash	(46min)
5---HWash	(63min)
6---DLY Wash	(46min)
7---DLY Hwash	(63min)
(DLY=1.5 hour start delay)	

Mode Selection v2.0 &2.1

1---Off	
1---Drain	(03.5min)
3---Rinse	(18.5min)
4---Wash	(55.0min)
5---HWash	(73.0min)

Front Panel indicators

When the Wash LED is lit, the unit is finished with initial rinses and has begun washing the dishes.

Adding dishes to the load at this time is NOT recommended.

If both Wash LED & Clean LED are lit, the second wash cycle of HWash is running.

If the Clean LED lit while the Wash LED is dark, the dishes are receiving their final rinse(s).

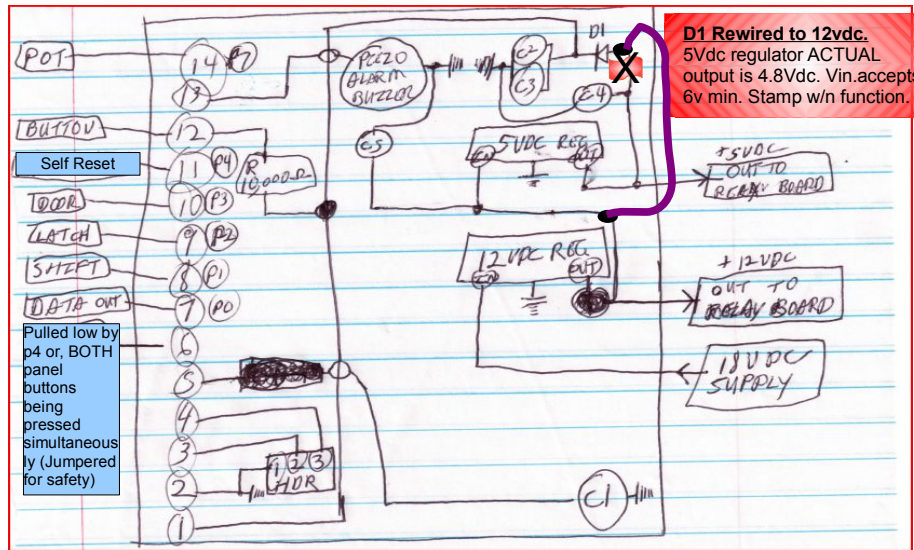
The Dry LED blinks slowly while washer is running. .

NOTE: The Dry LED stays on solid during water heating.

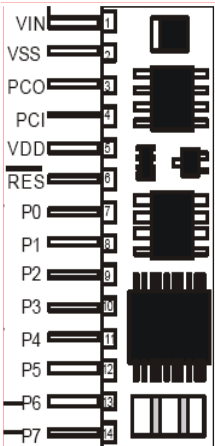
Stamp Board

Lines to Shift Board

SEENBY IN	COLORS
HEAD ON/OFF	DK BLUE
DATA	LT BLUE
SHEFT	LT BROWN
LATCH	DK BROWN
GROUND	DK GREEN
+5 VDC	LT GREEN
+12VDC	ORANGE



- C1----Tied to Stamp 05 (VDD) Internal regulator power filter cap. Avoids brownout from Pin activity.
- C2----Stamp Vin filter Cap #1.
- C3----Stamp Vin filter Cap #2.
- C4----5Vdc filter cap
- C5----12Vdc filter cap
- D1----Isolates C2&C3 > Stamp Vin from power fluctuations caused by relays/Leds.
- HDR-3 pin stamp programming hdr (Also, 2 Fxed VDC regulators 12v & 5v.)



BS1-IC Stamp Pins

- 01--Vin---Regulated 12VDC from C2/C3 passed by isolating D1. (Avoids Relay Drawdown brownouts)
- 02--VSS---Grounded & attached to programming header >0< installed on board.
- 03--PCO----Tied to programming header >1<
- 04--PCI----Tied to programming header >2<
- 05--VDD--Stamp internal regulator output Tied to filter cap C1. Avoids brownouts due to pin activity.
- 06--Reset-Pulled low by p4 or. BOTH panel buttons being pressed simultaneously (Jumpered for safety)
- 07--p0----Data out to Shift board (Lt. Blue wire)
- 08--p1----Serial data clock to shift Board. (Lt. Brown wire)
- 09--p2----Data memory latch clock to Shift board. (Dk. Brown wire)
- 10--p3----Door closed sensor. (Low w/closed)
- 11--p4----Tied to 06 thru jumper in parallel w/Panel button #2
- 12--p5----Panel "start" button. (Low w/pressed)
- 13--p6----Tied to 3-18Vdc low current piezoelectric buzzer alarm.
- 14--p7----Pot sensor for Heat temp.

PS2 Harness #1

- White----+12vdc
- Blue-----B0(wash)
- Green----B1(fill)
- Brown----B2(drain)
- Yellow----B6(soap)
- Red-----B7(heat)

Yellow Harness Items go to Stamp Board

All others to Shifter Board

PS2 Harness #2

- Purple----- (Broken)
- Green-----Ground
- Red-----+5vdc
- Blue-----B3(Wash)
- White-----B4(Dry)
- Orange-----B5(Clean)
- Black-----Door
- Yellow-----Thermistor
- Grey-----Button
- Brown-----Reset

Lt Orange-----12vdc
(External wire wrapped around Ps2 cable. Wire nut to board.)

Shift Board

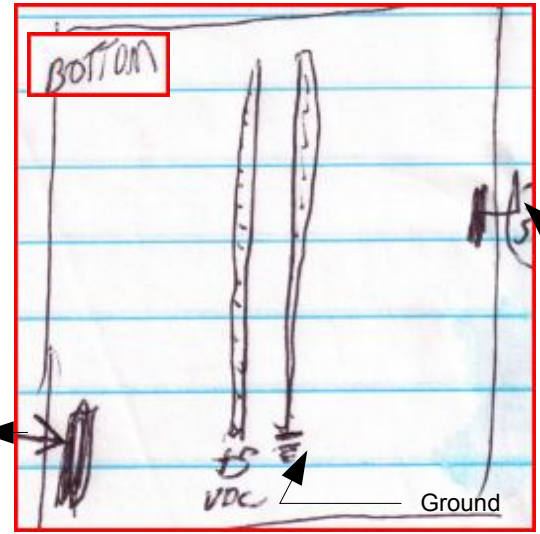
1/2

From Shift Board

SYMBOL IN	COLORS
HEAD ON/OFF SW	DK BLUE
DATA	LT BLUE
SHEFT	LT BROWN
LATCH	DK BROWN
GROUND	DK GREEN
+5 VDC	LT GREEN
+12VDC	ORANGE



Relays pulled from board & installed in relay harness.



Signals in from stamp board.

LEDs Out

IC1

Philips Semiconductors

Product specification

8-bit serial-in/serial or parallel-out shift register with output latches; 3-state

74HC/HCT595

IC2

ULN2803A
DARLINGTON TRANSISTOR ARRAY

- 500-mA Rated Collector Current (Single Output)
- High-Voltage Outputs . . . 50 V

IC1's 8 outputs Drive IC2's 8 lines as a buffer between relays & LEDs

Shift Bits
(Relay Harness)
0---Wash Rly
1---Fill Rly
2---Drain Rly
6---Soap Rly
7---Heat Rly

Shift Bits
(Front Panel)
3---Wash LED
4---Dry LED
5---Clean LED

PS2 Harness #1

White----+12vdc
Blue-----B0(wash)
Green----B1(fill)
Brown----B2(drain)
Yellow----B6(soap)
Red-----B7(heat)

Yellow Harness Items go to Stamp Board

All others to Shifter Board

PS2 Harness #2

Purple------(Broken)
Green-----Ground
Red-----+5vdc
Blue-----B3(Wash)
White-----B4(Dry)
Orange-----B5(Clean)
Black-----Door
Yellow-----Thermistor
Grey-----Button
Brown-----Reset

Lt Orange-----12vdc
(External wire wrapped around Ps2 cable. Wire nut to board.)

Shift Board

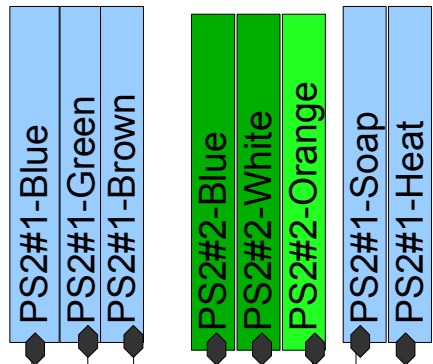
2/2

PS2 Harness #1

Blue-----B0(wash)
 Green----B1(fill)
 Brown---B2(drain)
 Yellow---B6(soap)
 Red-----B7(heat)

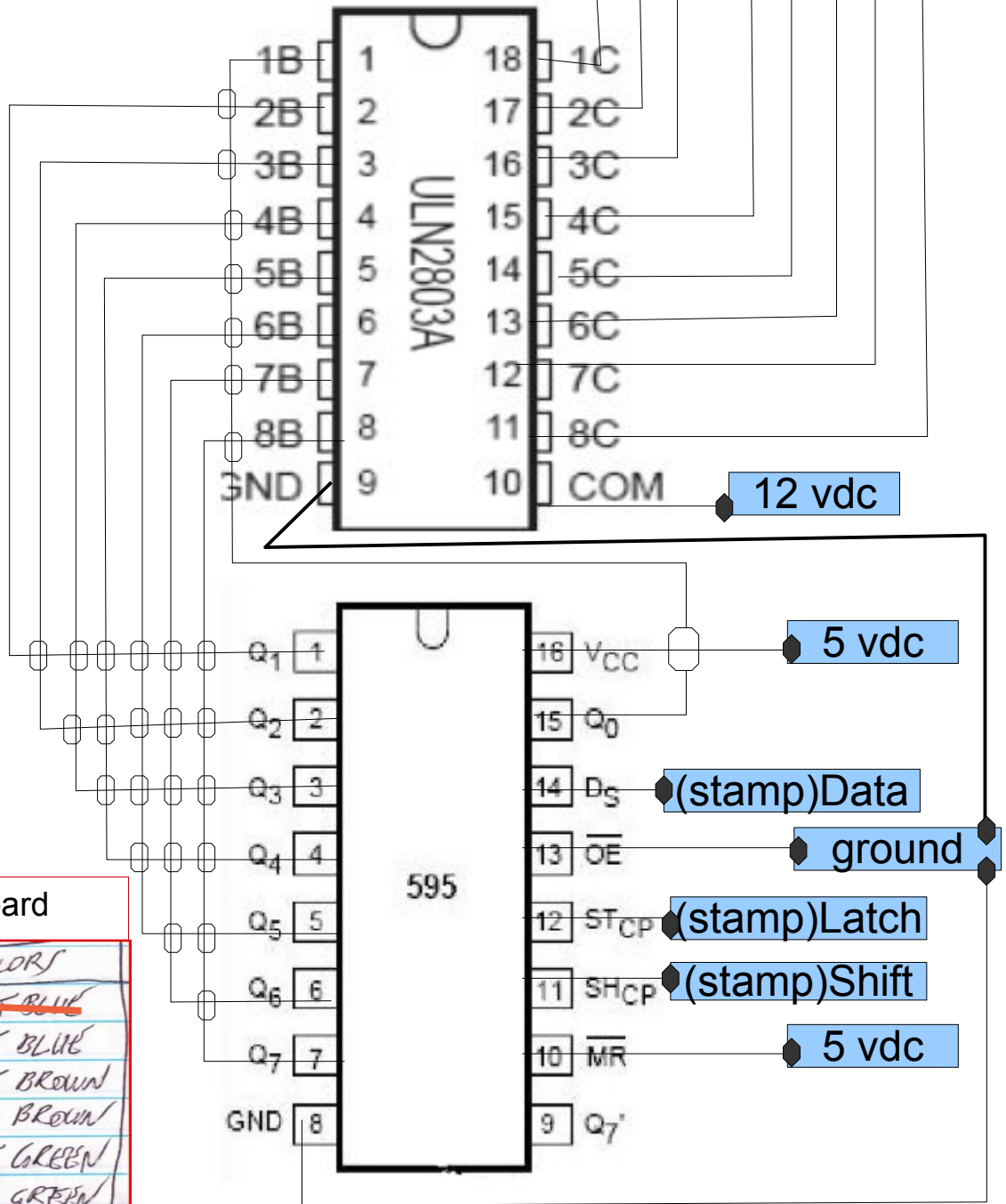
PS2 Harness #2

Blue-----B3(Wash)
 White-----B4(Dry)
 Orange-----B5(Clean)



Shift Bits
 (Relay Harness)
 0---Wash Rly
 1---Fill Rly
 2---Drain Rly
 6---Soap Rly
 7---Heat Rly

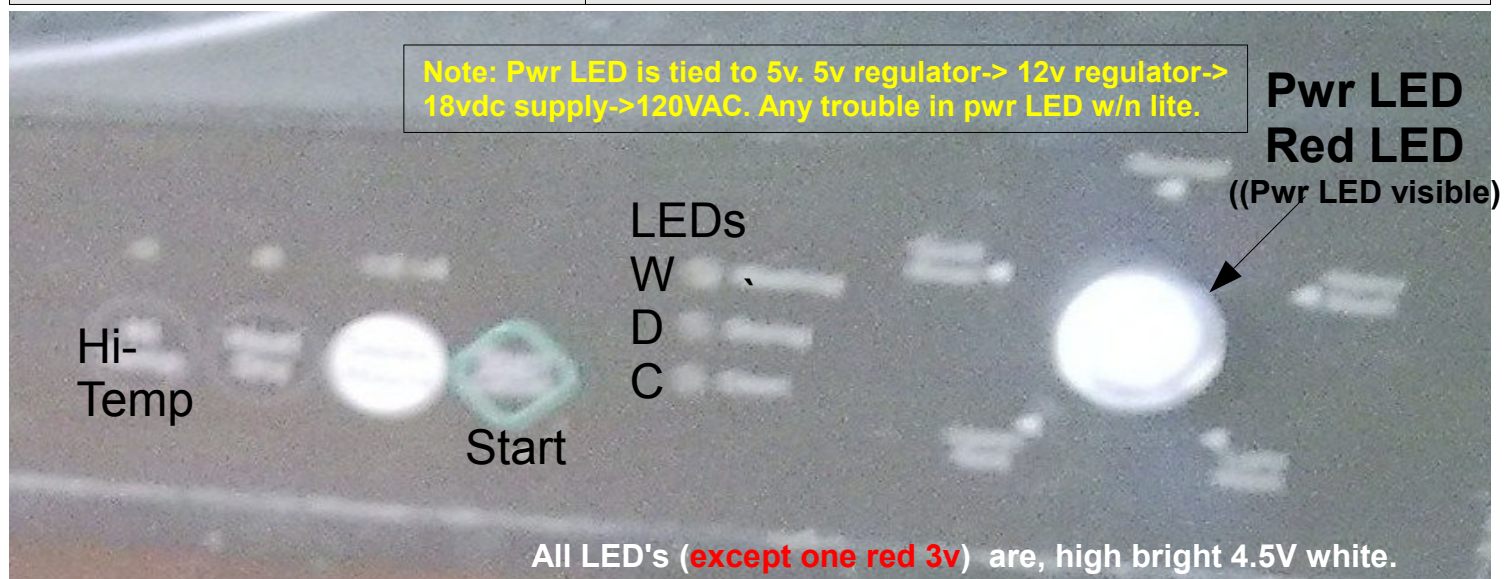
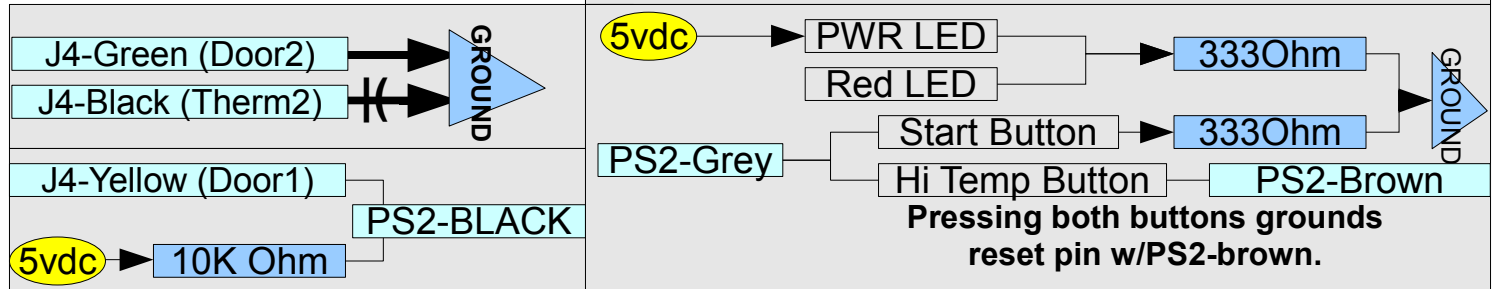
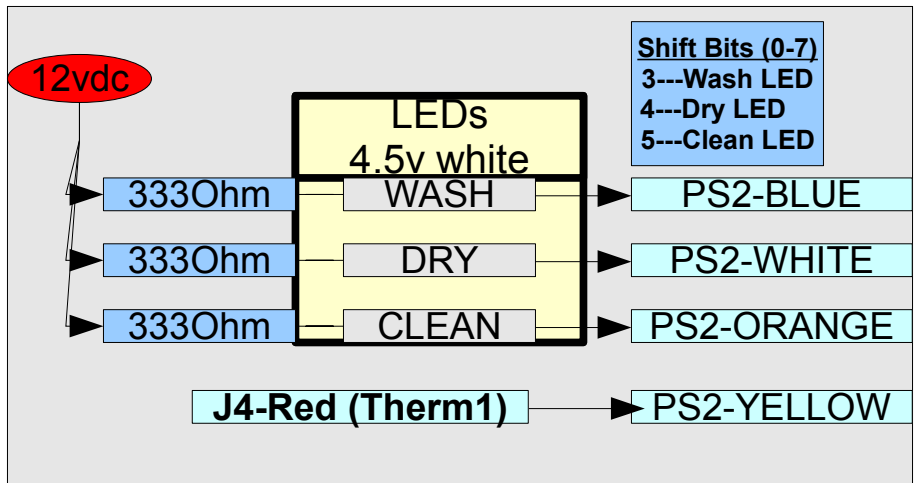
Shift Bits
 (Front Panel)
 3---Wash LED
 4---Dry LED
 5---Clean LED



Lines to Stamp Board

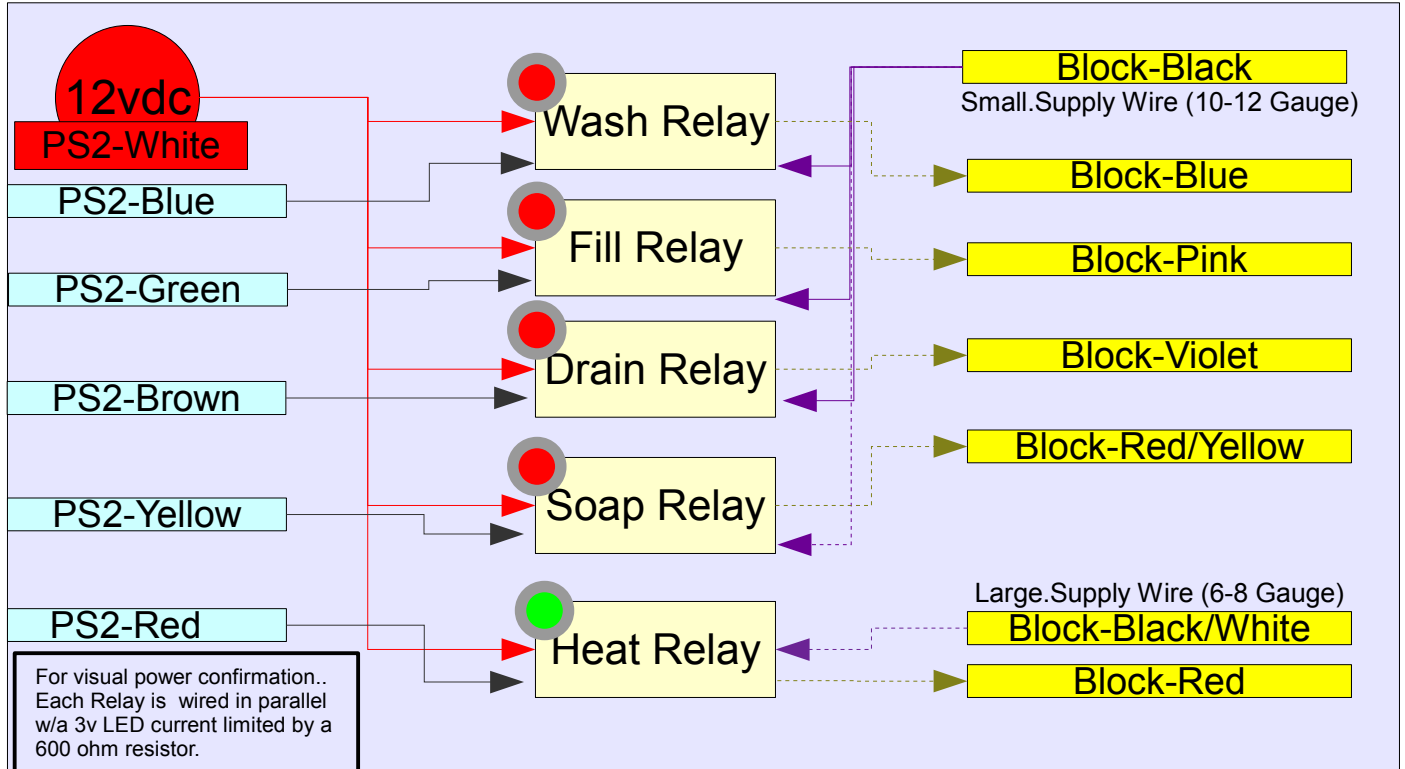
Stamp Board IN	COLORS
HEAT ON/OFF SW	DK BLUE
DATA	LT BLUE
SHIFT	LT BROWN
LATCH	DK BROWN
GROUND	DK GREEN
+5 VDC	LT GREEN
+12 VDC	ORANGE

Front Panel



J4 (to Relay Harness)	PS2 Harness #2
R--- Therm1	Purple------(Broken)
B--- Therm2	Green-----Ground
Y--- Door1	Red-----+5vdc
G--- Door2	Blue-----B3(Wash)
	White-----B4(Dry)
	Orange-----B5(Clean)
	Black-----Door
	Yellow-----Thermistor
	Grey-----Button
	Brown-----Reset
	Lt Orange-----12vdc
	(External wire wrapped around Ps2 cable. Wire nut to board.)
120VAC Door Sensor LED night lite driving a Photo resistor.	
Over 2Mohms (open) 1.5k ohms (closed)	

Relay Harness



PS2 Harness #1
 White----+12vdc
 Blue-----B1(wash)
 Green----B2(fill)
 Brown----B3(drain)
 Yellow----B7(soap)
 Red-----B8(heat)

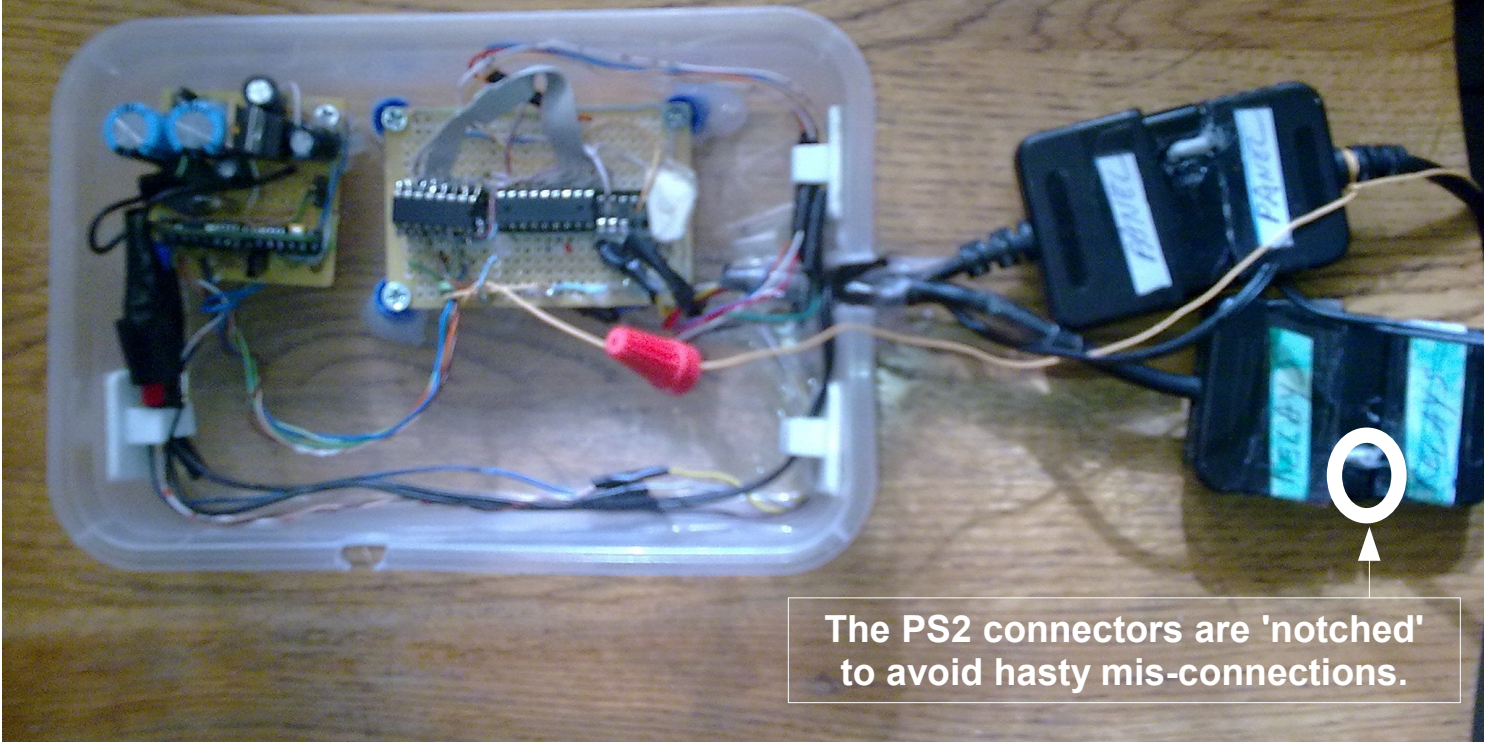
Shift Bits (0-7)
 0---Wash Rly
 1---Fill Rly
 2---Drain Rly
 6---Soap Rly
 7---Heat Rly

120VAC Block Connector
 Blue------(wash motor)
 Pink------(fill valve)
 Violet------(drain motor)
 Red/Yellow--(soap solenoid)
 Red------(heating element)
 Black------(110 unswitched)
 Black/white-(110 Door switched)

Note: The door's safety interlock switch disconnects both 120vac Hot & Neutral. Halts (overrides) all activity w/open door.

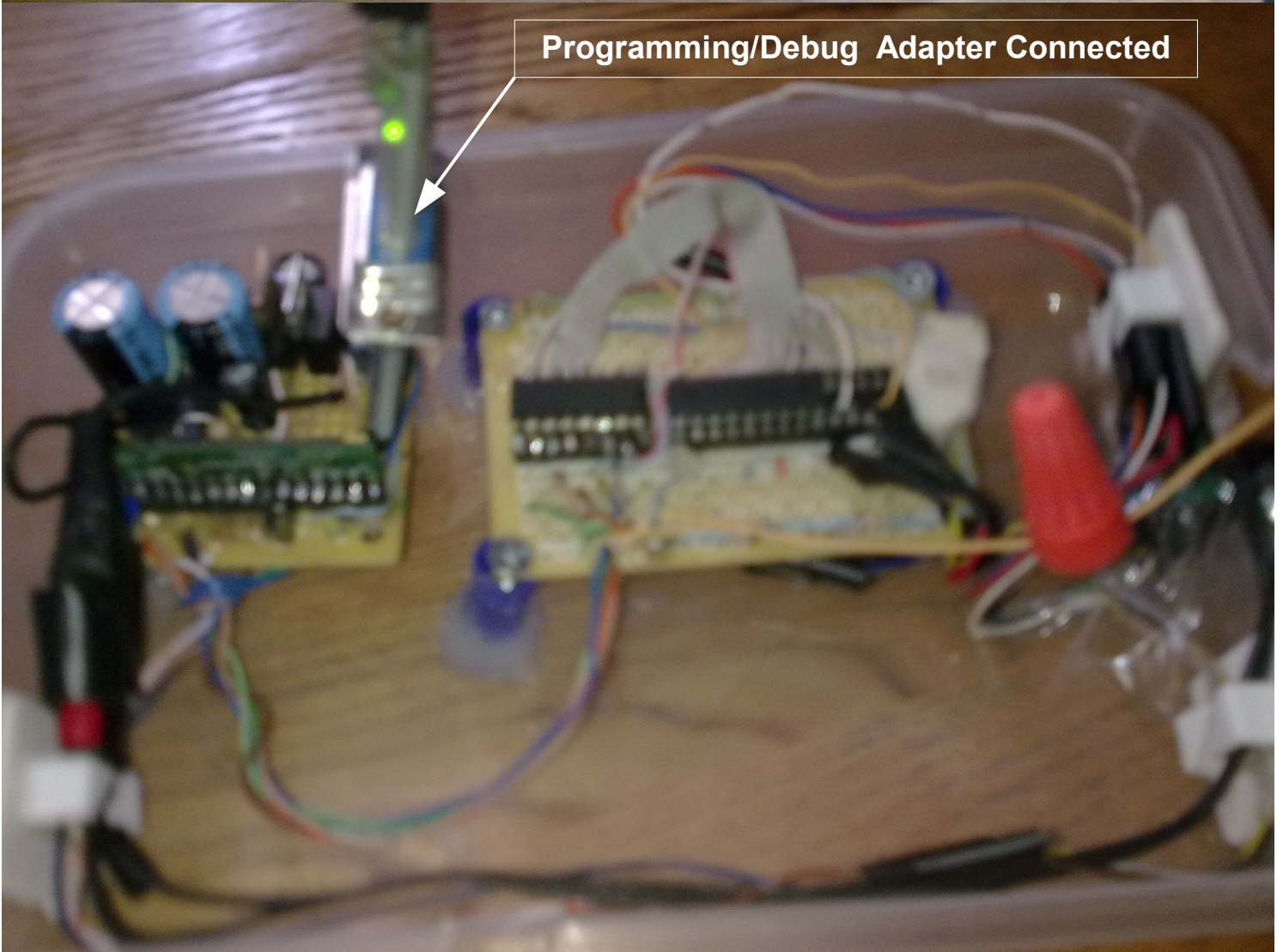
Images of Brain Housing

As the original brain unit was damaged by humidity, all openings are sealed.



The PS2 connectors are 'notched' to avoid hasty mis-connections.

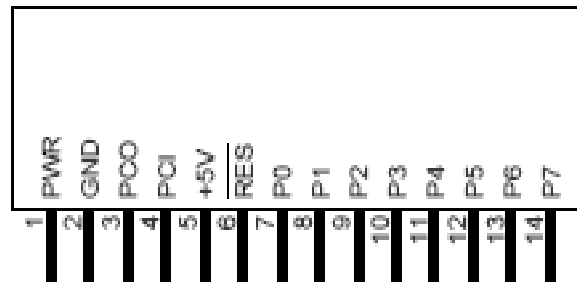
Programming/Debug Adapter Connected



BS1-IC

Relevant Data

BS1-IC



- PWR** **Unregulated power in:** accepts 6-15 VDC (6-40 VDC on BS1-IC rev. b), which is then regulated to 5 volts. May be left unconnected if 5 volts is applied to the +5V pin.
- GND** **System ground:** connects to PC parallel port pin 25 (GND) for programming.
- PCO** **PC Out:** connects to PC parallel port pin 11 (BUSY) for programming.
- PCI** **PC In:** connects to PC parallel port pin 2 (D0) for programming.
- +5V** **5-volt input/output:** if an unregulated voltage is applied to the **PWR** pin, then this pin will output 5 volts. If no voltage is applied to the **PWR** pin, then a regulated voltage between 4.5V and 5.5V should be applied to this pin.
- RES** **Reset input/output:** goes low when power supply is less than 4 volts, causing the BS1-IC to reset. Can be driven low to force a reset. Do not drive high.
- P0-P7** **General-purpose I/O pins:** each can sink 25 mA and source 20 mA. However, the total of all pins should not exceed 50 mA (sink) and 40 mA (source).

IC1

Relevant Data

Note:

Pin 10 wired High

Pin 13 wired Low

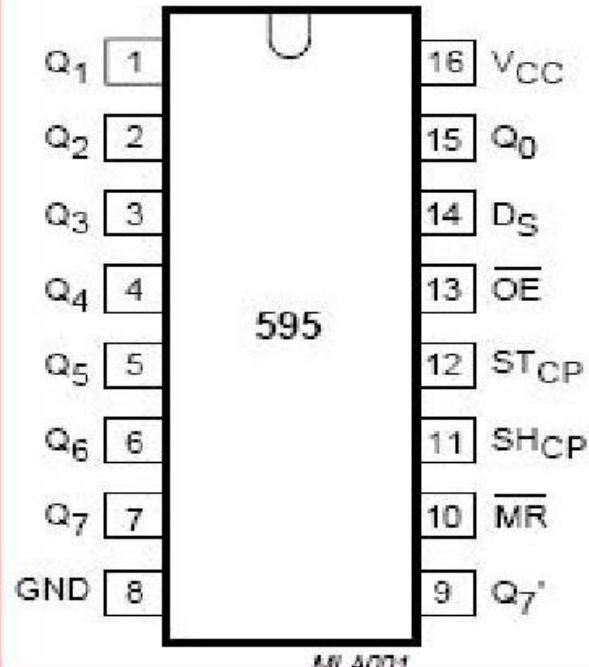
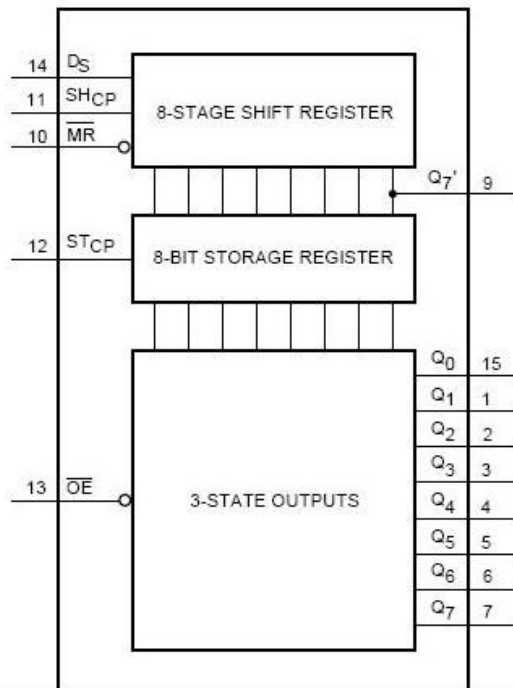
{All other inputs pulled High/Low by Stamp.}

Philips Semiconductors

Product specification

8-bit serial-in/serial or parallel-out shift register with output latches; 3-state

74HC/HCT595



PINNING

SYMBOL	PIN	DESCRIPTION
Q ₀ to Q ₇	15, 1 to 7	parallel data output
GND	8	ground (0 V)
Q ₇ '	9	serial data output
MR	10	master reset (active LOW)
SH _{CP}	11	shift register clock input
ST _{CP}	12	storage register clock input
OE	13	output enable (active LOW)
D _S	14	serial data input
V _{CC}	16	positive supply voltage

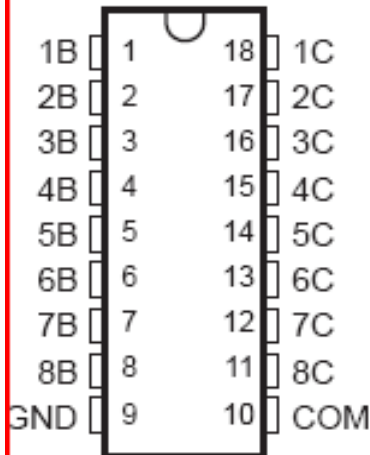
IC2

Relevant Data

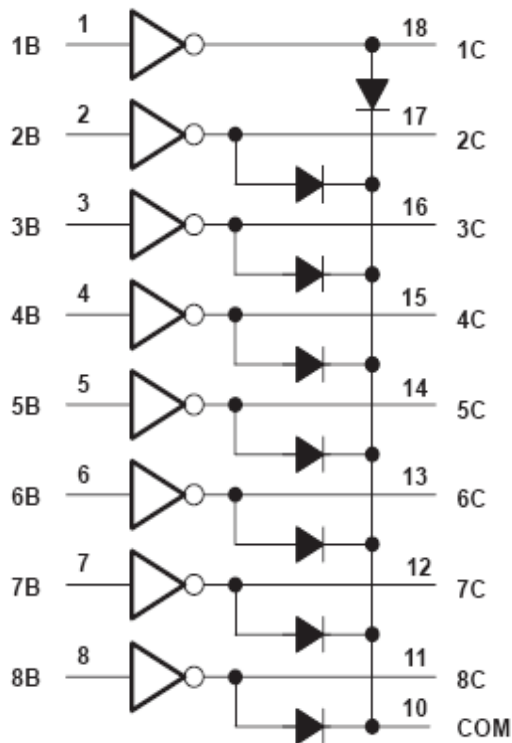
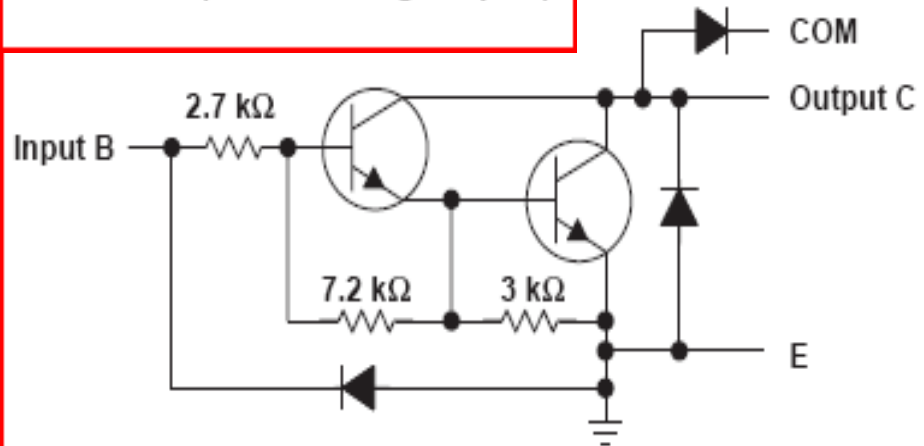
ULN2803A DARLINGTON TRANSISTOR ARRAY

- 500-mA Rated Collector Current (Single Output)
- High-Voltage Outputs . . . 50 V

DW OR N PACKAGE
(TOP VIEW)



schematic (each Darlington pair)



The ULN2803A is a high-voltage, high-current Darlington transistor array. The device consists of eight npn Darlington pairs that feature high-voltage outputs with common-cathode clamp diodes for switching inductive loads. The collector-current rating of each Darlington pair is 500 mA. The Darlington pairs may be connected in parallel for higher current capability.


```

*   uses debug.....try to avoid serial mouse error.....
*.....
*ver5 explores use of lookup table TO SAVE EPROM vs READ/WRITE
*   GAMMA items are most likely about to be dropped.
*.....
*ver6 using 2 minuteCnts instead of 1....this will shorten
*   data used by lookup tables...lots of repeat functs anyway.
*   2minute blocks will free eeprom space and be usefull.....
*.....
*ver7 using WashSubMode to branch to proper subsection
*   using less "if thens" & more eeprom saved
*   Note: 2 mins creates long wash cycles..perhaps 1.5 minutes
*.....
*Ver8 (working)implements button (low w/pressed) control.
*.....
*Ver9 (working)Moved timing Bytes to same word.Single word clear
*   Moved button bytes & mode bytes.clockreset no longer gosub
*.....
*Ver1.0 (RE-building)Rewire heat into ShiftDdata. P4 to reset IC
*   at end of each cycle. Removing need to reset ALL data
*   & re-init modes. (self-reset). Adding button#2 to panel
*   Wiring button#2 to high side of button#1. Must push both
*   buttons to drop reset low & manually reset washer.
*   SELF REST NOT WORKING INIT ALARM RINGS UNTILL MANUAL RESET
*.....
*VER1.0 Cleaned shifter code & it's Heater routines. Logic fixed
*   tweaked Pulseout timing in shifter. Retiming loopcntr
*.....
*ver1.3 Fixed bit logic error in wiring. Soap & Drain swapped.
*.....
*ver1.4 Fixed Heat "min" err. Caused it to jump to 255 from 1.
*.....
*ver1.5 fixed heat flickering on/off.
*   Added blinking Dryled (activity) during washer run.
*   Dled also goes steady when heating
*.....
*ver1.6 door still disabled...ground signal from door sensor low
*   current...must increase pullup resistor value.
*   Added timing adj pause during washmodes to DshiftC
*   Currently, pause 200 = .51 (+/-0.05) sec loops
*   90sec / .5 loops = 180 loops required
*   Re-configured wash modes to match timing of 45 minute wash
*.....
*ver1.7 Abandoned hope of eeprom savings by folding m1 into m2
*   Inspected code for BRANCH instead of IF. Saved Eeprom space
*   Installed M5&M6 (1.5 hour delayed start wash/heavy washes)
*.....
*ver1.8 Saved some more Eeprom by using one Dirs command instead
*   of various low & high commands to init outputs. Assuming
*   outputs are to be kept low instead.
*.....
*FUTURE Ver2.0 Settings=W0 Shiftdat=B0 Timing=B1 (clock minimal)
*   Accurate wash timings per cycle. using looped pause.
*   minute2cnt will become stepIdx
*   pause 4700 * 255(loopcntr) = 19.125 minutes
*   (May be longer, 255 loops adds interpeter delays .)
*   Will need to use Word instead of byte lookups.
*   Settings from initial cycle lookup must be stored.
*   Initial attempts indicate aggressive eeprom management.

```

```

*   Logic notes           Logic notes           Logic notes           Logic notes
*Logic overview (1 falls into 2 & etc unless specified.)
* The program flow DOESN'T exactly mirror the notes (mostly for troubleshooting)
* But, it DOES take the same logical paths.

```

```

'0--Init gosub g1 to zero out relay/leds) & Alert beep
'1--Loop start (first reset Settings' bits)
'  -----Use wash mode To BRANCH To: correct mode of operation --> 2,3,4,5,6,7,8
'2--m0-->BUTTON (button ONLY during m0)
'  -----Press & hold button till number of beeps matches desired Wash mode
'  -----beeps(1-5) 1=off & (2-7) = (m1-m8)
'  -----If no mode set goto 9 skipping clocks
'  -----When hold delay times out set the washmode,clear button data & goto 8
'3--m1-->Drain dishwasher (use LOOKUP & minute to set output bits)
'  -----If done...goto0 (self reset not working yet ..repeat alarm instead)
'  -----Goto8
'4--m2-->Rinse mode (use LOOKUP & minute to set output bits)
'  -----If mode NOT done goto 8
'  -----If washmode = rinse then goto0
'  -----Goto 5b
'5a-----Activate CleanLED archive bit..to be used by G1
'5b-----Advance WashSubMode by 1 & reset clocks & goto 1
'5a----->Wash mode (use LOOKUP & minute to set output bits)
'  -----If mode not done goto8
'  -----Goto 5a
'5--m3-->Wash mode Controller (WashSubMode branches TO 4,5b,4,0)
'6--m4-->Heavy Wash mode controller (WashSubMode branches TO 4,5b,5b,4,0)
'7--m5-->Self Reset Bsl (Take p4 LOW) <<<Not yet FUNCTIONAL>>>>
'8--Clocking operations -----If door open, Goto 9
'  -----(advance loopCnt+1)
'  -----If loopCnt > 1.5 mins THEN zero loopCnt & advance Minute2Cnt+1
'9--gosubG1
'A--gotol (loop program)
'
'-----gosubs-----gosubs-----gosubs-----gosubs-----gosubs-----gosubs-----
'G1--handles data shifting
'  Turns on heat based on heat value & Wrly & Drly. Cled is also set if Archive on
'  Also blinks Dled when not heating. If heating, Dled is steady on
'----->needed by init....hardware reset of 8 bit unit not in circuit.
'
'          End logic notes          End logic notes          End logic notes

```

'--->>>SETUP ENVIORNMENT

```

'->Variables
SYMBOL Settings = B0 'Planned to primary control w/bit(0-7) access. But...
' settings allows fast overall program access to all bits.
SYMBOL CntrlResrv = B1 'Reminder that bits 8-15 are reserved. B1 not usable
SYMBOL TmpWashMode= B2 'Tracks washmode being selected. Also blinks Dled during RUN.
SYMBOL OnButthldCnt=B3 'used to count delay b4 TmpWashMode is locked in.
SYMBOL WaterTemp = B4 'WaterTemp value from POT of Pin 7
SYMBOL HeatTime = B5 'Used to lock Heat on as needed to avoid abusing relay.
SYMBOL Wmode = B6 'Chooses "drain,rinse,wash,hvywash" startup modes
SYMBOL WashSubMode= B7 'used to hop thru wash/rinse modes during washes
SYMBOL Minute2Cnt = B8 'used to count 1.5 minutes elapsed per step
SYMBOL loopCnt = B9 'used in loop to count time for 90 seconds.
' used:shifting,debug & pause to create about .5 seconds/loop
SYMBOL X = B11 'Garbage temp variable. & For...Next loop counter
' (NOTE: GOSUB shift routine uses X twice)
SYMBOL ResetCmds = W1 'Used to reset both Button Cmds at once. saves Epron
SYMBOL ResetHeat = W2 'Used to reset both Heat bytes at once.
SYMBOL ResetWmode = W3 'Used to reset both Wmode bytes at once.
SYMBOL ClockReset = W4 'Used to reset both clock bytes at once.
'Note: The bytes assigned to each Word are assigned so that they worktogether
' Although, it turns out, only 2-3 were required.

```

```

'shifter bits shifting *2 for MSB out [divde by 2 FOR LSB out]
'circuit wired for MSB operation
'some of these bits not to be used...keeping for reference
SYMBOL Wrly = BIT0 '-- 1 read boolean by shifter to Trigger heater.
'SYMBOL Frly = BIT1 -- 2
SYMBOL Drly = BIT2 '-- 4 read boolean by shifter to Cancel heater.

```

```
'SYMBOL Wled      = BIT3  -- 8
SYMBOL Dled      = BIT4  '-- 16  Flashed during Run/Steady on during Heat.
'SYMBOL Cled      = BIT5  -- 32
'SYMBOL Srly     = BIT6  -- 64
SYMBOL Hrly     = BIT7  '--128  read boolean by shifter to trigger Dled.
```

```
'-bits 8-15 slated for control data.
```

```
SYMBOL CleanLED  = BIT11 'used boolean to light bit5 during final rinse.
```

```
'->constants
```

```
'this section relates to the bit's associated with the relays & LEDs
```

```
SYMBOL F          = 2  'fill      --- relay
SYMBOL FW         = 3  'fill-wash --- relays
SYMBOL FS         = 66 'fill-soap --- relays (Rinse Agent-dispenser)
SYMBOL W          = 1  'wash      --- relay
SYMBOL WD         = 5  'wash-drain --- relays
SYMBOL D          = 4  'drain     --- relay
SYMBOL waSW       = 73 'Soap-wash & wash LED (Wash suggested to flush soap out)
SYMBOL waF        = 10 'Fill     & wash LED
SYMBOL waFW       = 11 'Fill-wash & wash LED
SYMBOL waW        = 9  'wash     & wash LED
SYMBOL waWD       = 13 'wash-drain & wash LED
SYMBOL waD        = 12 'Drain    & wash LED
```

```
'the following constants make the code easy to read.
```

```
SYMBOL OffMde     = 0
SYMBOL ModeDone   = 0
SYMBOL Active     = 1
SYMBOL DrainMode  = 1
SYMBOL RinseMode  = 2
SYMBOL WashMode   = 3
SYMBOL HvyWashMode = 4
SYMBOL Heat       = 128
```

```
'->SETUP PINS
```

```
SYMBOL DatOut     = PIN0
SYMBOL Door       = PIN3
SYMBOL OnButton   = PIN5
```

```
DIRS = %01000111
```

```
'LOW 0      ' PIN0  =Data out
'HIGH 1     ' PIN1  =Shift clock out
'HIGH 2     ' PIN2  =Storage clock out
            ' PIN3  =Door open signal in
            ' PIN4  =Goes Low to reset BS1 (input allows float)
            '---note:DO NOT take PIN4 high...prevents programming.
            ' PIN5  =Button in      (low when pressed)
'LOW 6     ' PIN6  =Buzzer out
            ' PIN7  =Pot tester for thermistor in
```

```
'=====
'==>>> Program <<<==
'=====
```

```
Init:
```

```
GOSUB dshift
```

```
FOR x =1 TO 20
```

```
  PULSOUT 6, 4000  'alert beep
```

```
NEXT
```

```
MAINloop:
```

```
'=====
'=== washer mode logic ===
'=====
```

```
'branch m1-m4 (1-drains water>>2-rinses only>>3&4-are wash modes )
```

```
Settings= OffMde 'ALWAYS reset relay/led data (each loop)
```

```
POT 7,1,WaterTemp
```

```
BRANCH Wmode, (m0,m1,m2,m3,m4)
```



```

'Special modes M5/M6/M9 are here out of sequence for ease of locating.
'm5---1.5 hour delay Wash
'm6---1.5 hour delay Heavy Wash
m5:
m6:
IF Minute2Cnt < 60 THEN clocks
Wmode = Wmode - 2
ClockReset = 0
GOTO mainloop

'M9 -----Self-Reset
m9:
LOW 4
' -----END of out of sequence modes-----

'M0 -----Button command interpretation
m0: ' Accepts commands from Button to select mode
    ' (To stop washer, Reset will be used to restart to init.)

BRANCH OnButton, (Bpress, BnotPress)
BnotPress:
BRANCH OnButtHldCnt, (Skiptime) 'button wasn't pressed yet, nothing to do.
    OnButtHldCnt = OnButtHldCnt + 1 'Count up to unit start.
    IF OnButtHldCnt < 6 THEN skiptime 'not done waiting for person to leave buttons alone
    Wmode = TmpWashMode 'set The mode...unit will start next loop.
GOTO Skiptime

Bpress:
    OnButtHldCnt = 1 'start count, button pressed. we WILL BE SETTING A MODE.
    TmpWashMode = TmpWashMode + 1 // 5 '5 beeps (1-Off 2-Drain 3-Rinse 4-Wash 5-HvyWash)
    FOR x= 0 TO tmpWashMode 'STARTS W/2 BEEPS... LOOPS BACK AROUND TO ONE.
        PULSOUT 6, 10000 'ALLOWING YOU TO CANCEL START IF NEED BE.
        PAUSE 150
    NEXT
GOTO Skiptime

'M1 -----Drain only
m1: 'Drain
LOOKUP Minute2Cnt, (d,d),Settings
BRANCH Settings, (INIT)
'If Minute2Cnt > than 1 it won't change Settings from zero = mode done so init.
'This technique is used in all following modes
GOTO Clocks

'M2 -----Rinse mode
m2: '15 minute cycle
LOOKUP Minute2Cnt, (f,fw,fw,fw,wd,f,fw,wd,d,d),Settings
IF Settings > ModeDone THEN Clocks
IF Wmode = RinseMode THEN INIT 'No need to flip into washmode selector
GOTO m301 'Washmode m3 or m4 enabled...send to selector system

'M3 (&M4)-----Wash Mode directors
m3:
BRANCH WashSubMode, (m2,m302,m2,INIT) 'submode is increased after each mode completes
m4:
BRANCH WashSubMode, (m2,m302,m302,m2,INIT)
'Using another branch for m4. Less epron than if..then handling

'm300-----Set clean LED from here on..
m300:
CleanLED=1 'then fall into 301

'm301-----wash modes selector (changer)
m301:
WashSubMode = WashSubMode + 1 'Advance to next step & reset clocks
ClockReset = offMde

```

GOTO MAINloop

'M302 -----Wash mode (runs 2x for heavy wash)
m302: 'Wash-16.5 minute cycle (triggers soap @ end for Rinse agent dispenser
LOOKUP Minute2Cnt, (waF,waFW,waSW,waFW,waFW,waFW,waFW,waFW,waWD,waD,FS),Settings
BRANCH settings,(m300) 'if settings =0 then set Clean LED & set next step.
'fall into Clocks

'Clocks-----TimeKeeping

Clocks:

'do time counting stuff b4 shift

BRANCH door,(skiptime) 'door open--pause clock.

loopCnt= loopCnt + 1 // 181 'configured after testing for about 90 seconds time.
'must change when adjusting shifting timing

IF loopCnt <> 180 THEN Skiptime 'time hasn't passed (should be one less than max)
Minute2Cnt= Minute2Cnt+ 1

SkipTime:

GOSUB dshift 'update output

GOTO MAINloop 'Main Loop end....do it again.

'=====

'===== gosubs =====

'=====

Dshift:' Requires Settings to be "prep"ed b4 call
' used as a gosub to allow "near instant" software reset by init.
' because hardware reset of 8 bit shift unit not in circuit.
' Added routines to control Dled,Cled & Heat

DshiftA:

IF Wrly=0 OR Drly = 1 THEN DshiftC ' if not washing or if draining...no heat.
x=0

LOOKUP WaterTemp,(0,0,0,6,5,4), x '(0-2 discarded as static.)

'No specifics, so guessed Cap value...picking up static from wash motor <110F

HeatTime = HeatTime + x MIN 1 'These ignore faulty low values &

HeatTime = Heattime - 1 MIN 1 'avoids relay abuse from flickering on/off.

BRANCH HeatTime,(DshiftC,DshiftC) 'MIN 0 can cause byte rollover to 255. 1 is safer

settings = settings + 144 'turn on Hrly & Dled

DshiftC:

LOOKUP CleanLED, (0,32), x 'If Cled archive bit set then Light Cled.

settings = settings + x

BRANCH Wmode,(DshiftD) ' Don't adjust/use the following if not running.

OnButtHldCnt = OnButtHldCnt + 1 // 5 'used to set freq of Dled blink

LOOKUP OnButtHldCnt,(1),Dled 'Dled lights on zero for .5 sec

PAUSE 200 ' Tweak to make loops to be approx .5 secs

DshiftD:

DEBUG %Settings 'Dump packet to Serial Output.

DshiftE:

FOR x=1 TO 8

DatOut = BIT7 'pin0 get's data to go out

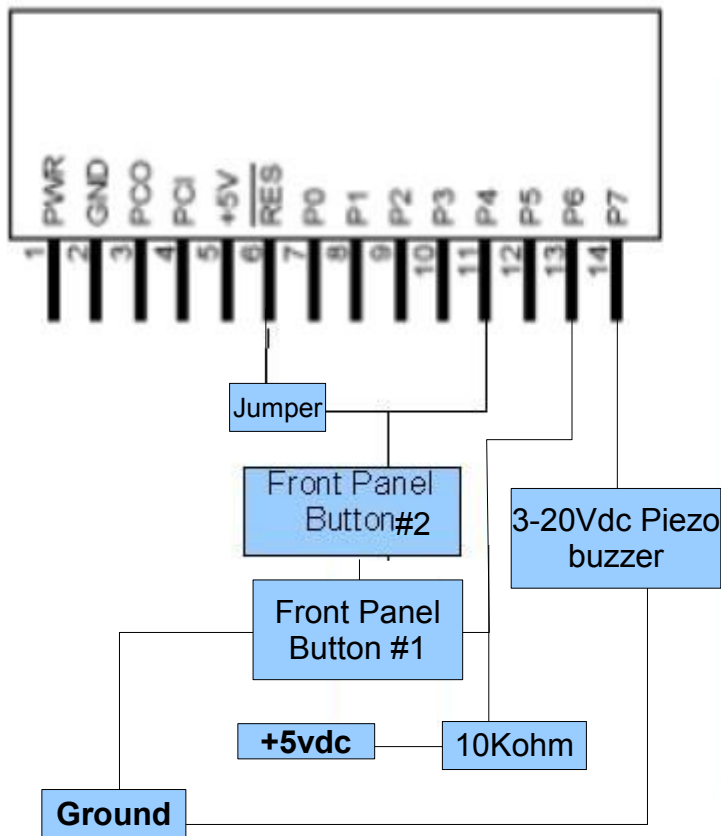
PULSOUT 1 ,8 'clock to shift data into register

Settings = Settings * 2 'Shift data MSB first. (right 2 left)

NEXT

PULSOUT 2 ,8 'shift complete...latch data to outputs

RETURN



```

SYMBOL OnButton = PIN6
SYMBOL x = B2
LOW 7

Init:
  FOR x =1 TO 20
    PULSOUT 7, 4000 'alert beep
  NEXT

Hold:
  BRANCH OnButton,(m5,hold)

m5: 'Self reset
  Low 4
  input 4
  goto m5

```

Mike you're right. As I posted that's exactly what I would expect. Although (as you knew), I'd used a slightly more expanded circuit for testing.. it included a buzzer that sounds on prog init. **I'm attaching the more appropriate circuit &** **Apologizing for rushing off that response post.** When I run this setup...the bs1...doesn't reset unless I press both buttons.