# Senior Design II – Fall 2004

# Swim Team

# Event and Heat Display

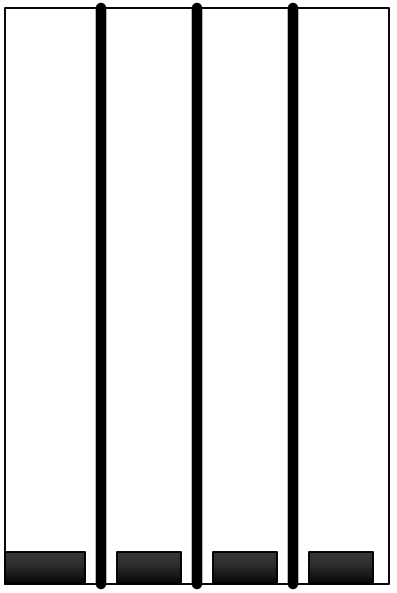
# Final Presentation

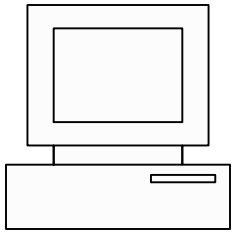
Sang-Hoon Choi Nate Keane Karl South

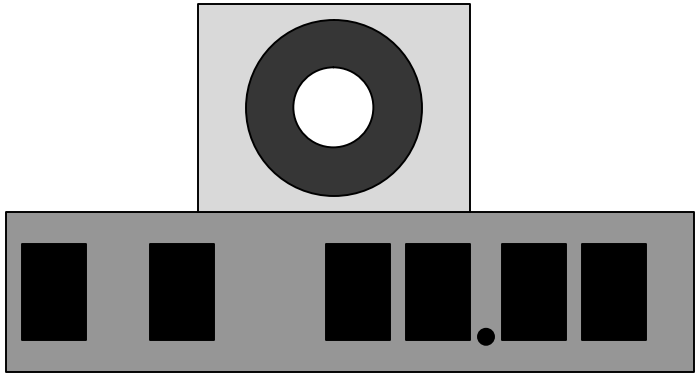
# Project Background

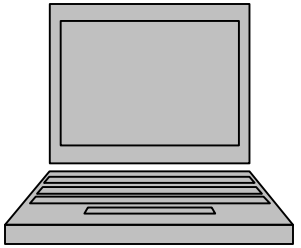
Swimming Team Scoreboard Project

- Inland Empire Swimming Committee is a group of 19 swimming clubs.
- They all share a single scoring system.
- Scoring system consists of:
  - Timing computer
  - Scoring computer running special “Meet Manager” software
  - Touchpads for the swimmers
  - Single line scoreboard









# Project Background

## Problem

- Timing Computer can control up to 12 single line scoreboards.
- IESC only has 1.
- Cycle through a variety of information:
  - **Current elapsed running time**
  - **Lead split time (if event requires more than 1 lap)**
  - **Back to the running time**
  - **The final times for each lane**
  - **Instead of team scores, IESC displays the Event and Heat number.**

# Project Background

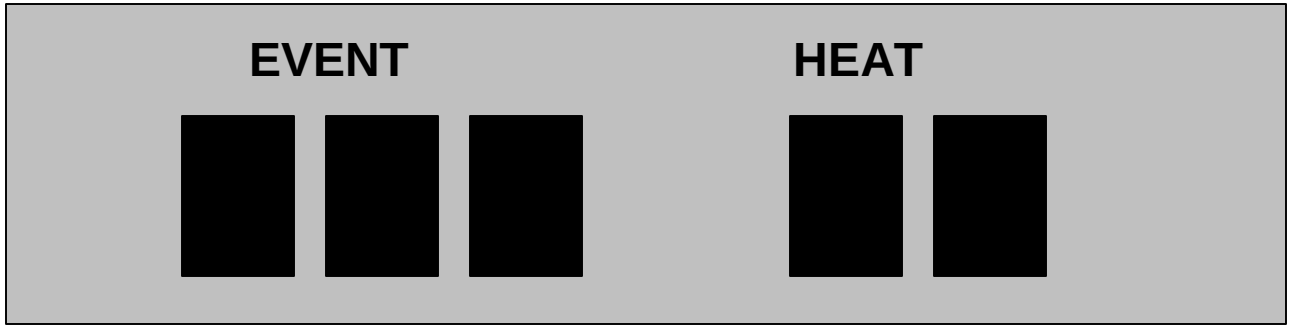
## Problem cont.

- Most meets have over 100 swimmers.
- The event & heat number are displayed for 3 seconds.
- This makes it hard for parents & swimmers to know what the current event is.
- This results in swimmers missing their events.

# Project Background

## Project Goal

- Build a scoreboard to display the current event and heat number.



# Objectives

1. Data Retrieval & Interpretation
   - Intercept data coming from the timing computer.
   - Extract Heat & Event information.

2. Data Transmission
   - Hardlink from the existing scoreboard.

3. Display the Data
   - Maxim 7 segment display driver.
   - Large 7 segment displays.

# Project Breakdown

## Nate - Data Retrieval & Interpretation

- Intercept data coming from the timing computer.
- Extract event and heat information.

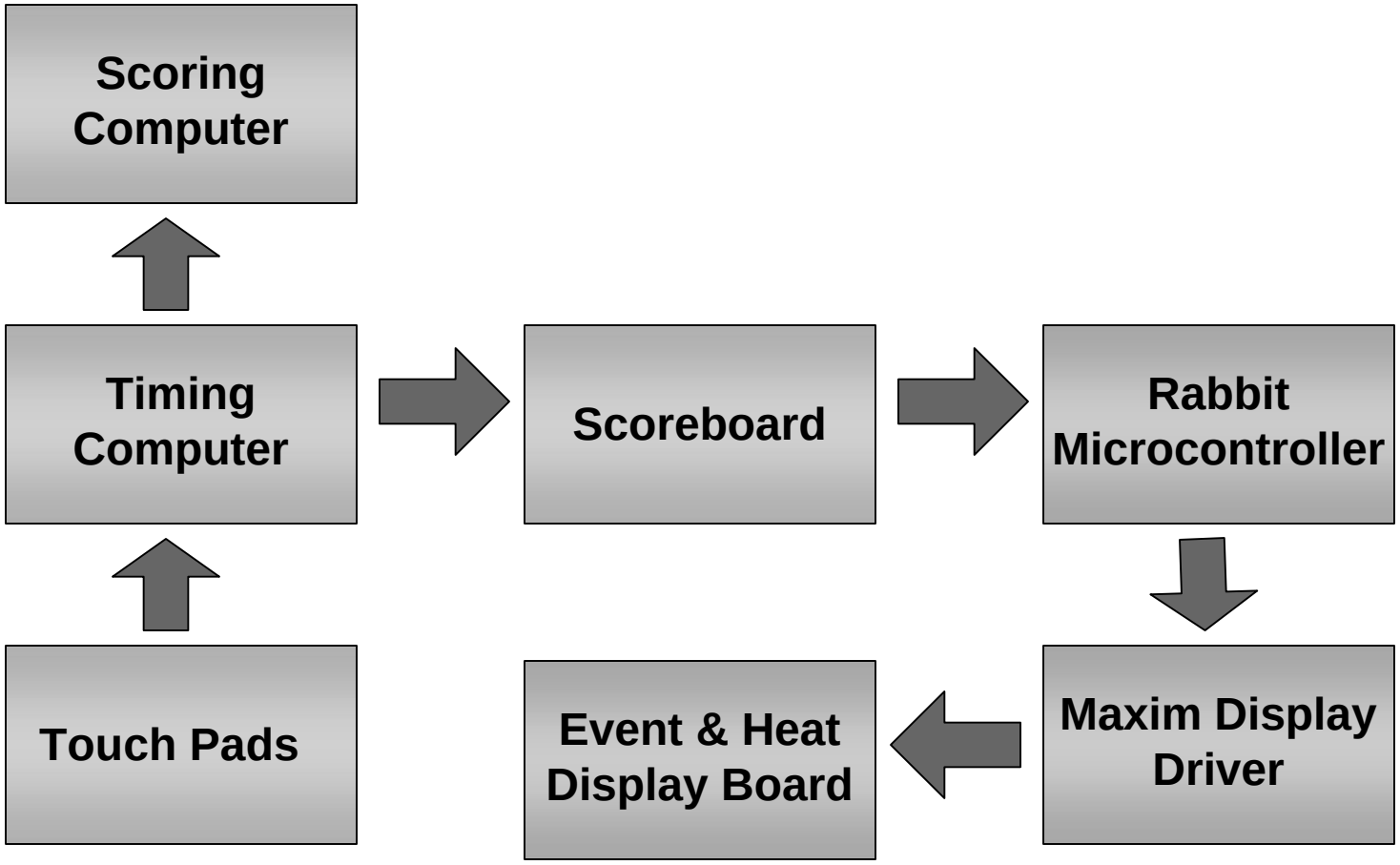
## Choi - Data Display

- Control Maxim display driver.
- Send event & heat to Maxim.

## Karl - Data Display

- 7 segment display operation.
- Power Supplies.

# Main Data Path Overview



# Data Retrieval & Interpretation

## - Timing Computer Operation

• Timing computer sends data to the scoreboard using RS232 style serial communications.

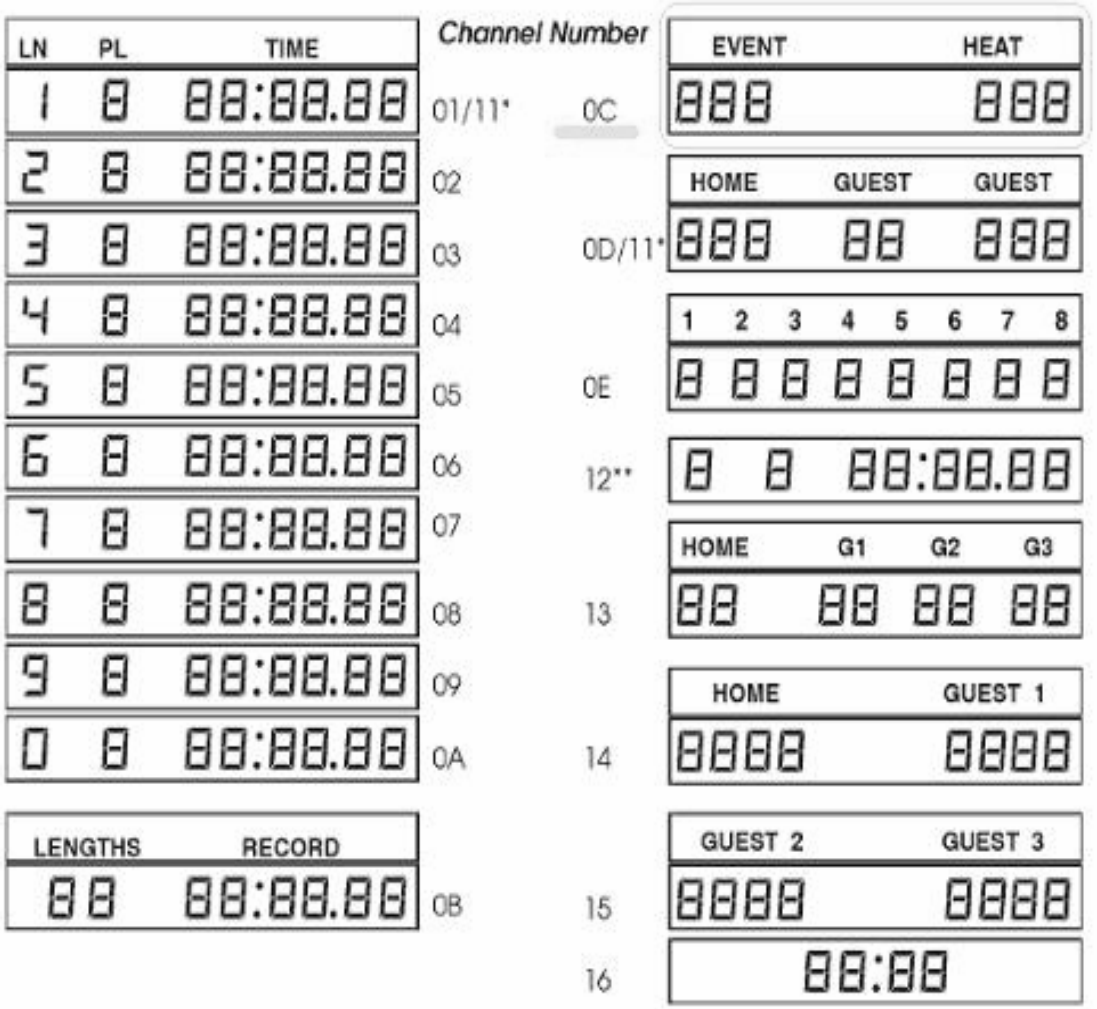
• Uses ¼” phono plugs and wires to transmit the data.

• The existing scoreboard has a “phono out” that we will tap into.

# Data Retrieval & Interpretation

## - Timing Computer Operation

- The timing computer sends out 19 channels of information.
- Channel's 1 – 0A, display the split, and final times for each lane.
- Channel 0C sends out the event and heat number.
- Channel 0B displays record information.
- The individual scoreboards have switches to control which channel they intercept.



# Data Retrieval & Interpretation

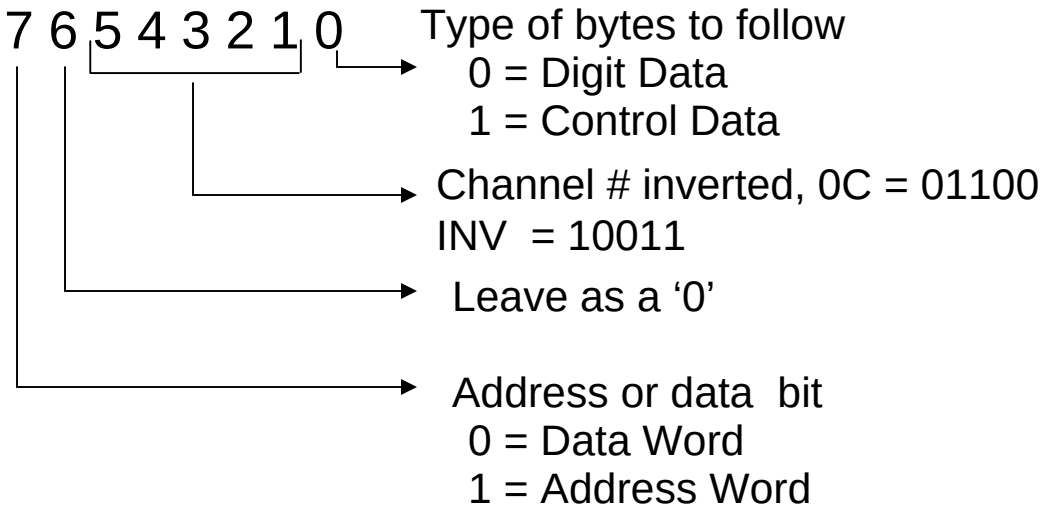
## - Timing Computer Operation - How does it send the data?

• Data is transmitted at 9600 BAUD, 8 data bits, 1 stop bit, with even parity.

- Transmits 3 types of bytes:

Type 1. Address Byte

- Contains the channel number, defining which scoreboard should respond.



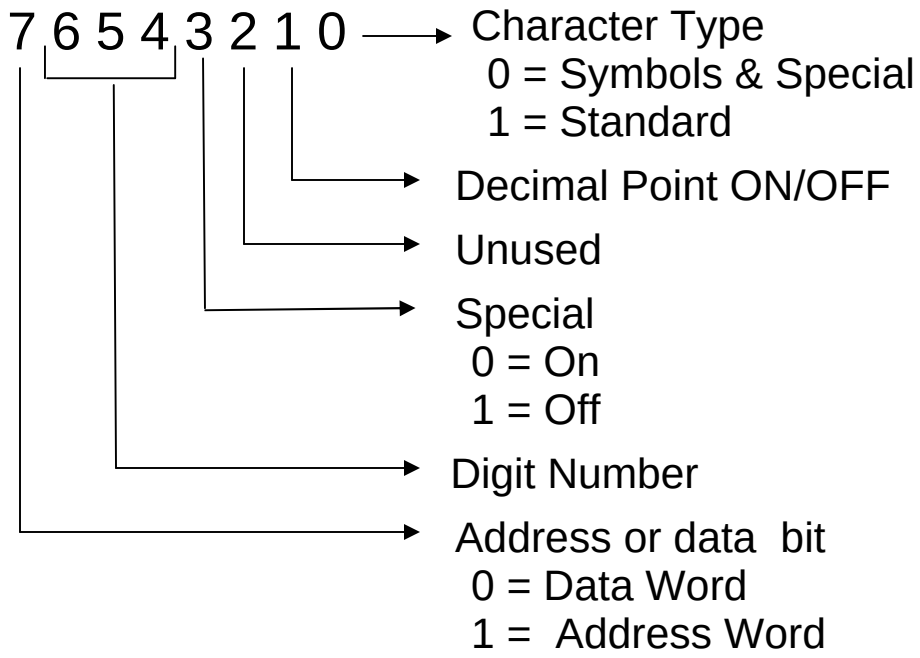
# Data Retrieval & Interpretation

## - Timing Computer Operation - How does it send the data?

- Transmits 3 types of bytes:

  Type 2. Control Byte

  - Chooses regular or special characters, decimal points, etc.



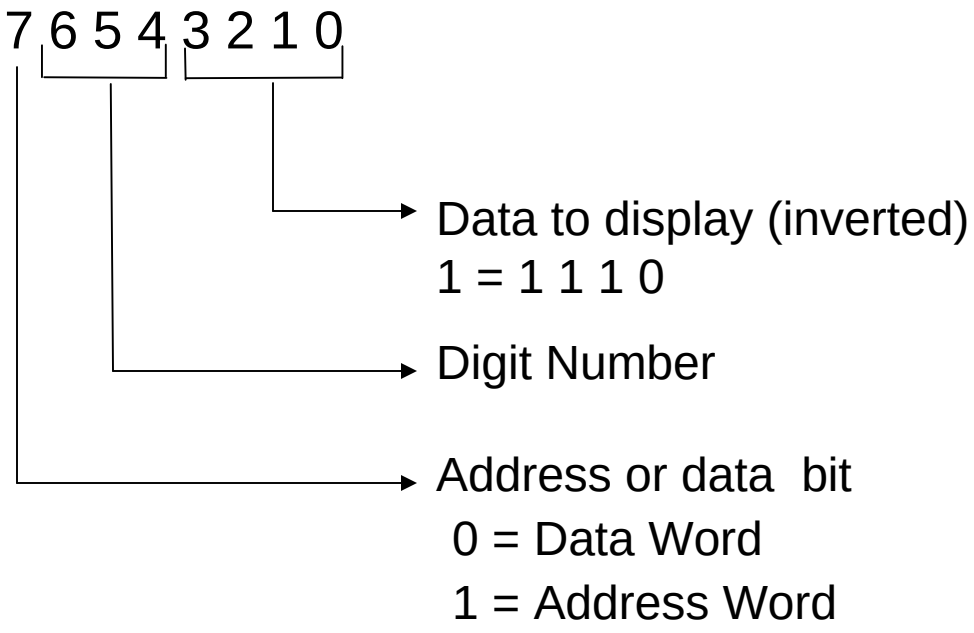
# Data Retrieval & Interpretation

## - Timing Computer Operation - How does it send the data?

- Transmits 3 types of bytes:

  Type 3. Data Byte

  - Tells what numeral to display & which digit to display it on.

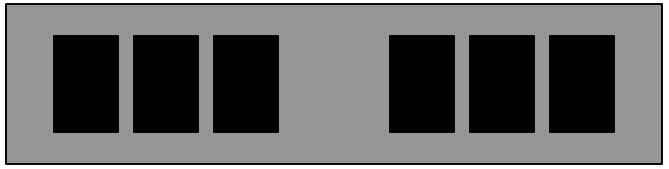


# Data Retrieval & Interpretation

## - Timing Computer Operation - How does it send the data?

- Datastream from Timing Computer consists of 2 digit hex values.

- Example: Display Event 1, Heat 2.

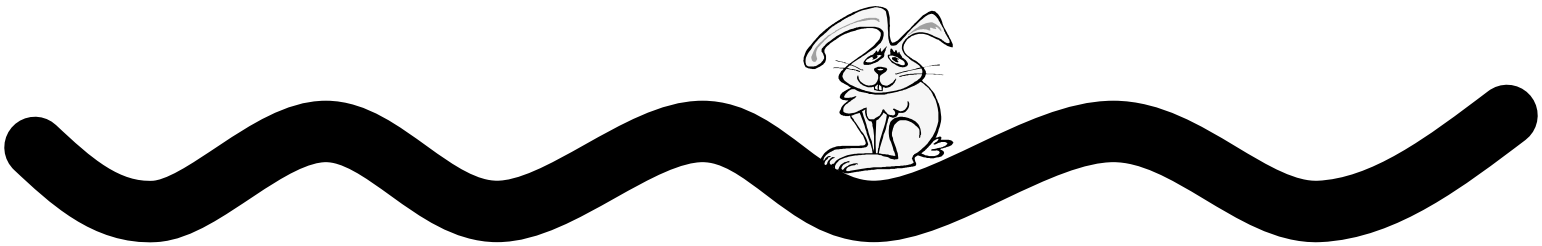


| | |
|---|---|
| 1. Address Byte (1 0 10011 1) – Channel 0C, Control Data Follows | 0xA7 |
| 2. Control Bytes (0 000 1 1 1 1) – Digit 0, special characters off, decimal off, standard symbols & characters. | 0x0F |
| • repeat for digits 1 - 5 | 0x1F<br>...<br>0x5F |
| 3. Address Byte (1 0 10011 0) – Channel 0C, digit data follows. | 0xA6 |
| 4. Data Byte (0 000 0000) – Data byte, digit 0, display a 0 (no inverting for 0's). | 0x00 |
| • Digit 1, display a 0 | 0x10 |
| • Digit 2, display a 1 (0 010 1110) data inverted | 0x2E |
| • Digit 3, (first digit of the heat number) | 0x30 |
| • Digit 4, (second digit of the heat number) | 0x40 |
| • Digit 5, display a 2 (0 101 1101) data inverted | 0x5D |

# Data Retrieval & Interpretation

## - Event & Heat Extraction

- Rabbit microcontroller intercepts the data from the datastream.

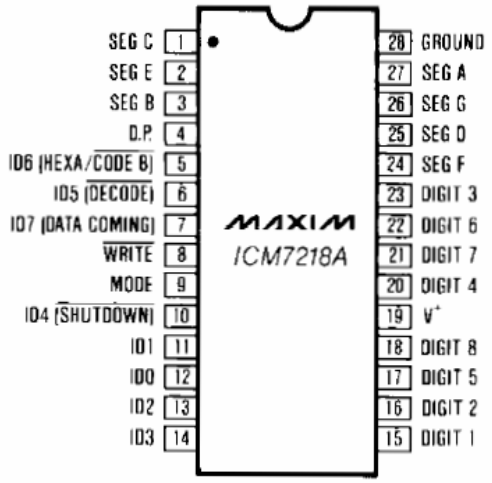


- Ignores values except 0xA6 (channel 0C).
- 0xA6 signals that the next bytes will be event and heat data.
- When 0xA6 is received, Rabbit stores the next bytes until a new channel number is received (hex value greater than 0x7F).
- If a new channel number is received before 8 bytes have been stored, the event and heat were interrupted and stored data is discarded.
- Otherwise, the stored data is sent to the data interpreter function.
- Data interpreter decodes the event and heat information, first 3 bytes are the event, last 3 bytes are the heat number, and updates the event_num and heat_num variables.

Which are used by Choi’s send to Maxim function.

# Data Display

- Control Maxim Display Driver  <Maxim driver>




- Maxim ICM 7221A
- interfaces microprocessor to an 8 digit 7-segment displays
- accept data in a serial format and drive displays

- Maxim display driver has 10 inputs and 16 outputs
- inputs : Mode, write, and ID0 – ID7
- output : Digit 0 – Digit 8, SEG A – SEG G and D.P

# Data Display

- Control Maxim Display Driver  < Input >

| INPUT | PIN | STATE | FUNCTION |
|---|---|---|---|
| WRITE | 8 | High<br>Low | Input Not Loaded Into Memory<br>Input Loaded Into Memory |
| MODE | 9 | High<br>Low | Loads Control Word on WR<br>Loads Input Data on WR |
| ID0–ID2,<br>DIGIT ADDRESS | 12, 11, 13 | High<br>Low | Loads "one"<br>Loads "zero" |
| ID3, BANK SELECT | 14 | High<br>Low | Select RAM Bank A (Hex or Code B<br>Select RAM Bank B Data only) |
| ID4, SHUTDOWN<br>(MODE High) | 10 | High<br>Low | Normal Operation<br>Shutdown |
| ID5, DECODE/NO DECODE<br>(MODE High) | 6 | High<br>Low | No Decode<br>Decode |
| ID6, HEX/CODE B<br>(MODE High) | 5 | High<br>Low | Hexadecimal Decoding<br>Code B Decoding |
| ID7, DATA COMING<br>(MODE High) | 7 | High<br>Low | Data Coming (control word)<br>No Data Coming (control word) |
| ID0–ID7, INPUT DATA<br>(MODE Low) | 5–7, 10–14 | High<br>Low | Loads "one" (Note 1)<br>Loads "zero" (Note 1) |

-Control Register (When Mode is high and Write is low)

- ID3 : Bank select
- ID4 : Shutdown
- ID5 : Decode/No Decode
- ID6 : Hex/Code B
- ID7 : Data Coming

# Data Display

- Control Maxim Display Driver  <Address and Data>

| DATA LINES | | | Selected |
|---|---|---|---|
| ID2 | ID1 | ID0 | Digit |
| 0 | 0 | 0 | D1 |
| 0 | 0 | 1 | D2 |
| 0 | 1 | 0 | D3 |
| 0 | 1 | 1 | D4 |
| 1 | 0 | 0 | D5 |
| 1 | 0 | 1 | D6 |
| 1 | 1 | 0 | D7 |
| 1 | 1 | 1 | D8 |

- **ADDRESS**
  - When Mode is high and Write is low
  - Address Digit : ID0 – ID2
  - Microprocessor sends them with control registers (ID3 – ID7)

# Data Display

## - Control Maxim Display Driver <Address and Data>

| ID3 | ID2 | ID1 | ID0 | HEXADECIMAL | CODE B |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 2 | 2 |
| 0 | 0 | 1 | 1 | 3 | 3 |
| 0 | 1 | 0 | 0 | 4 | 4 |
| 0 | 1 | 0 | 1 | 5 | 5 |
| 0 | 1 | 1 | 0 | 6 | 6 |
| 0 | 1 | 1 | 1 | 7 | 7 |
| 1 | 0 | 0 | 0 | 8 | 8 |
| 1 | 0 | 0 | 1 | 9 | 9 |
| 1 | 0 | 1 | 0 | A | - |
| 1 | 0 | 1 | 1 | b | E |
| 1 | 1 | 0 | 0 | C | H |
| 1 | 1 | 0 | 1 | d | L |
| 1 | 1 | 1 | 0 | E | P |
| 1 | 1 | 1 | 1 | F | (Blank) |

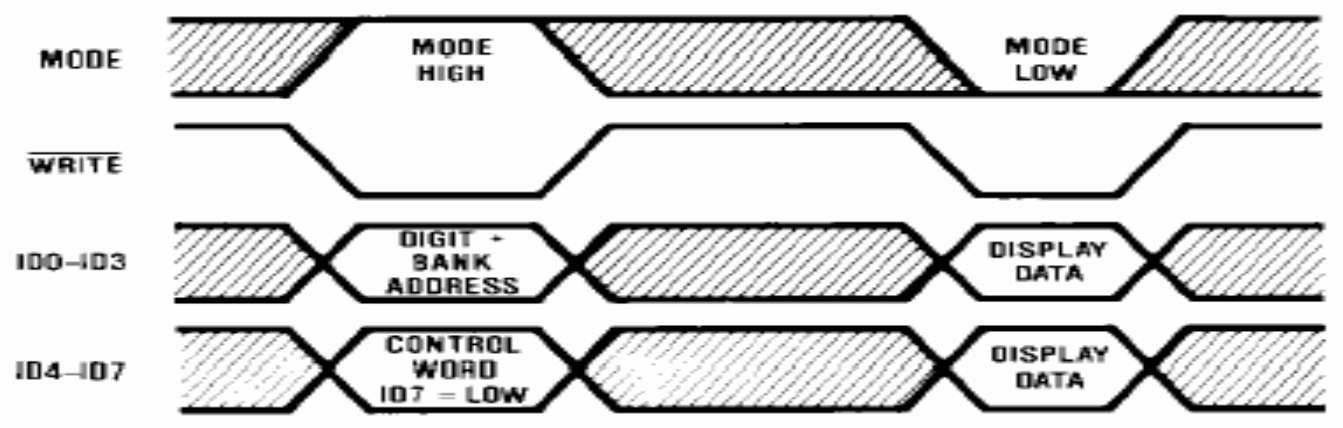
-**DATA**

- When Mode is low and Write is low
- Data Digit : ID0 – ID7
- ID0 – ID3 are used
- ID4 – ID7 are don’t care bits
- To display number we can use both Hex and

  Code B mode

# Data Display

- Control Maxim Display Driver <Timing Diagram>




-Single Digit Updates Mode

- One digit to be changed without updating the entire display.
- Whenever heat and event number are changed, 5 digits will be updated respectively
- First, the control register is updated with Mode high, Data coming low, and the address of the digit to be updated
- Second, transfer the data from ID0-ID7 into the selected digit's RAM location with Mode low.

# Data Display

## - Connections from Rabbit to Maxim

-Rabbit microprocessor Needs 10 output pins. Will use, 2 from port C, 3 from port D, and 5 from port E.

→ **Connections from Rabbit to Maxim**

- Mode = pin9 on display driver, PD3 on rabbit
- WRITE = pin8 on display driver, PD4 on rabbit
- ID0 = pin12 on display driver, PD5 on rabbit (LSB of display address and data)
- ID1 = pin11 on display driver, PE0 on rabbit (middle bit of addr and data)
- ID2 = pin13 on display driver, PE1 on rabbit (MSB of display address)
- ID3 = pin14 on driver, PC0 on rabbit, data (1 RAM Bank A, 0 RAM Bank B)
- ID4 = pin10 on driver, PE2 on rabbit, shutdown (1 normal, 0 shutdown)
- ID5 = pin6 on driver, PC2 on rabbit (data)
- ID6 = pin5 on driver, PE4 on rabbit, (1=hex, 0=code B, use code B)
- ID7 = pin7 on driver, PE5 on rabbit, (1=data coming, 0 = no data coming)

# Data Display

## - Maxim Display Driver Operation - How does it send the data?

- Sending data from Maxim driver to 7-segment display

- Example: Display Event 001, Heat 02.

1. When send address and control registers

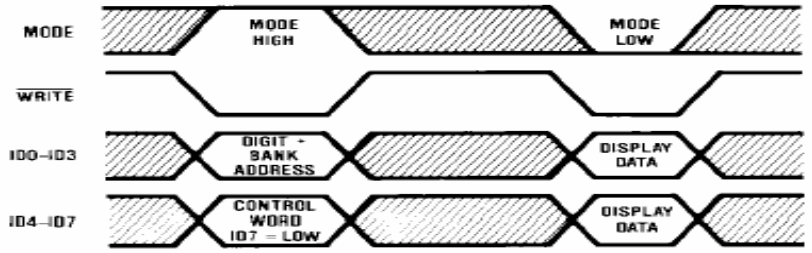
   Mode high, Write low, ID7 low(no data coming), ID6 low(Code B), ID5 low(nodecode)

   ID4 high(no shutdown), ID3 high(Select RAM bank A), and ID2-ID0 address digits

2. When send data

Mode low, Write low, and ID0 – ID7

(ID4 – ID7 : Don’t care)




EVENT3 : Address is 000(digit 1), Data is 0000. <**1000011000**, **00xxxx0000**>

EVENT2 : Address is 001(digit 2), Data is 0000. <**1000011001**, **00xxxx0000**>

EVENT1 : Address is 010(digit 2), Data is 0001. <**1000011010**, **00xxxx0001**>

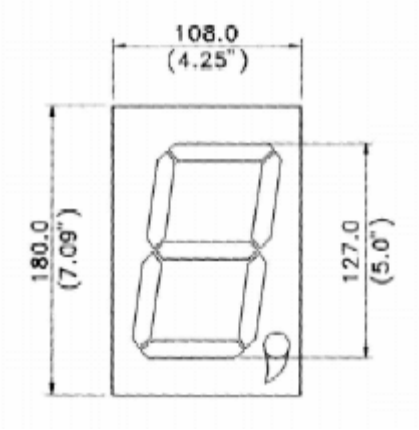
HEAT2 : Address is 011(digit 4), Data is 0000. <**1000011011**, **00xxxx0000**>

HEAT1 : Address is 100(digit 5), Data is 0010. <**1000011100**, **00xxxx0010**>

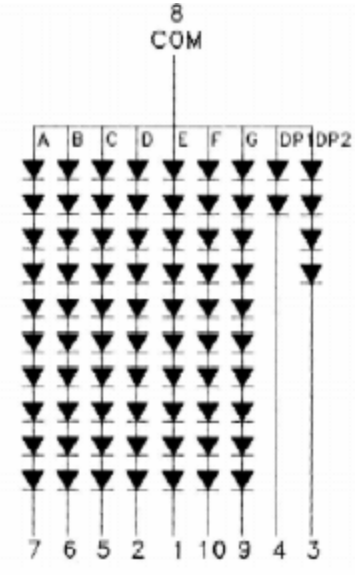
And then We come to Karl’s physical display…

# 7-Segment Digits

- Dimensions
  - Digit Height 5 Inches
  - Segment Height 2.5 Inches

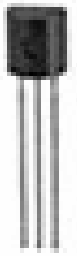



- Common Anode Type
  - Each Segment is Made of 10 Diodes in Series
  - Forward Voltage Per LED is 2 Volts @ 20mA
  - Wavelength Per LED 660 nm
  - Luminous Intensity Per LED is 90 Milli-Candela (mcd)



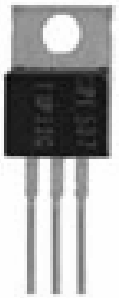
# MOSFET Transistors

- N Channel MOSFET
  - 2N7000 TO-92 Package
  - Drain-Source Voltage 60 V
  - Drain-Gate Voltage 60 V
  - Gate-Source Voltage 40 V (pulsed)
  - Drain Current 500 mA (pulsed)



TO-92 Package

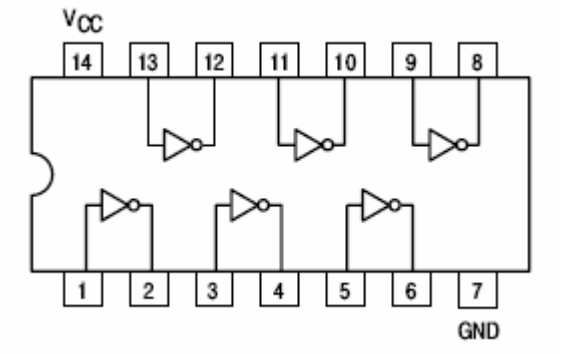
- P Channel MOSFET
  - SPP08P06P TO-220 Package
  - Drain-Source Voltage -60 V
  - Gate-Source Voltage 20 V
  - Drain Current -35 A (pulsed)



TO-220 Package

# Logic Inverter and Power Supply

- SN7404 Logic Inverter
  - VCC is 5 Volts
  - Input High > 2.0 Volts
  - Input Low < 0.8 Volts




- 24 Volt Power Supply
  - Vin is 240/120 VAC
  - Vout is 24 VDC
  - Output Current  is 5 Amps

# System Voltages

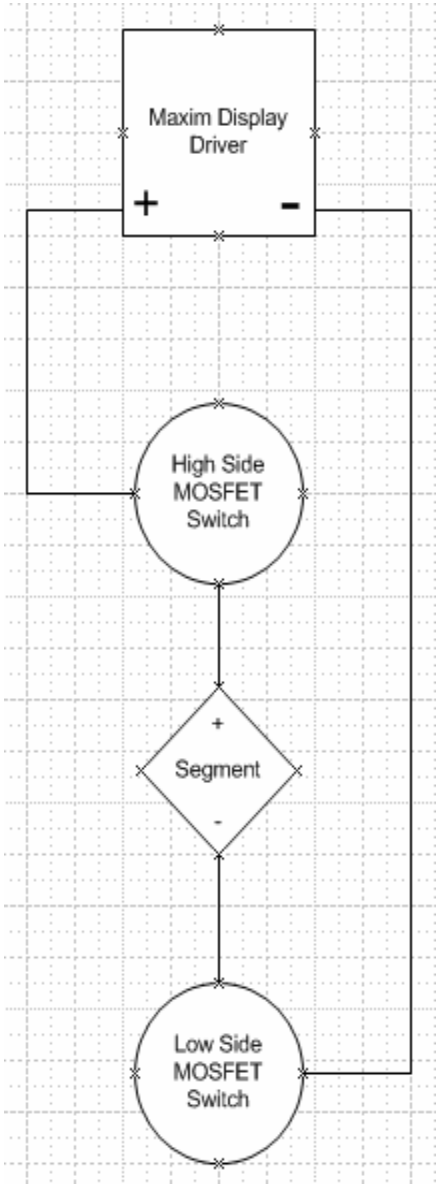
| | |
|---|---|
| Rabbit Microcontroller | 12 Volts |
| Maxim Display Driver | 5 Volts |
| Logic Inverter | 5 Volts |
| MOSFET High Side Switch | 24 Volts |
| MOSFET Control Signal from ICM7218A | 0-5 Volts |

# Maxim Display Driver Output

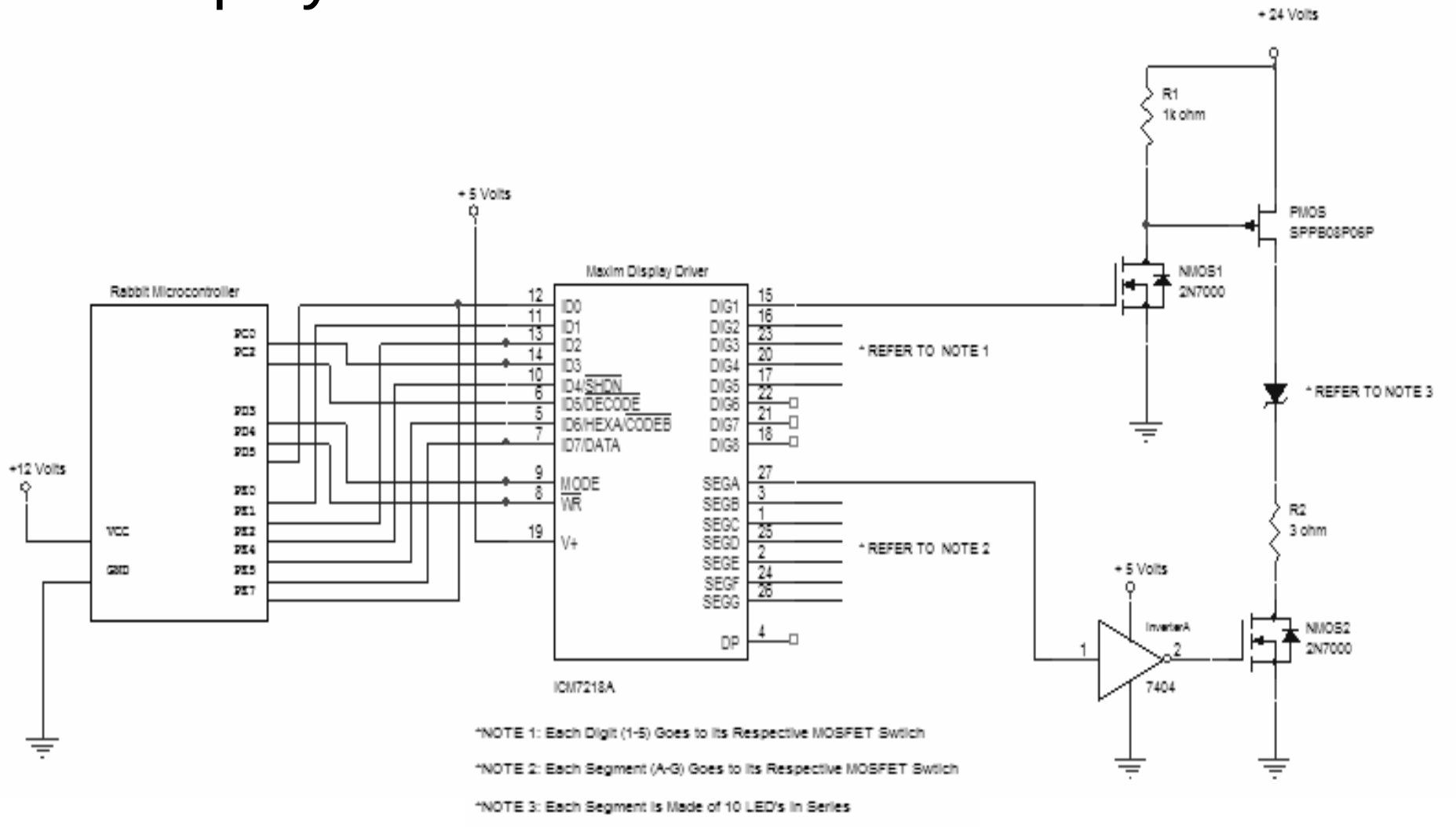
- + is Maxim Display Driver Digit Output
- - is Maxim Display Driver Segment Output

**Maxim Output Truth Table**

| Maxim Output Signal | + | - | Segment Status |
|---|---|---|---|
| | 0 | 0 | Off |
| | 0 | 1 | Off |
| | 1 | 0 | On |
| | 1 | 1 | Off |



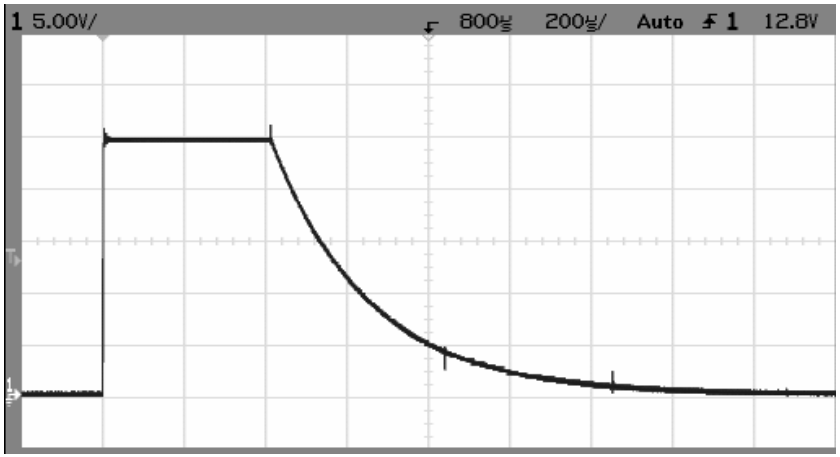
# Display Driver Schematic



# MOSFET High Side Switch Time Constant

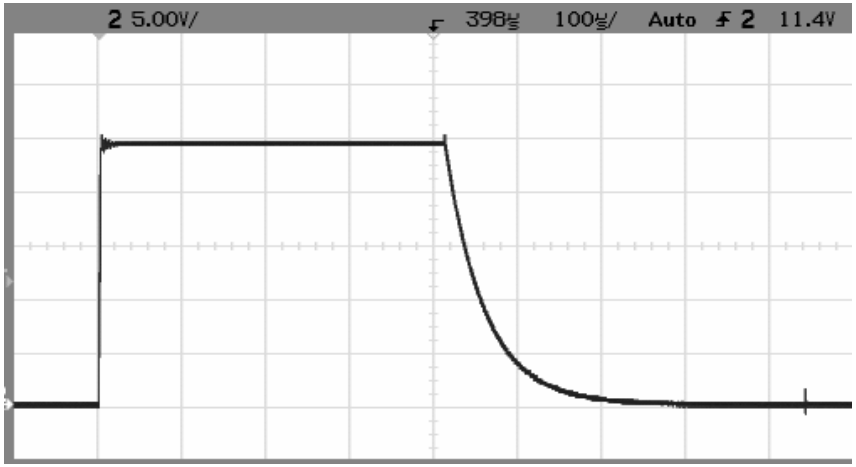
5K Resistor

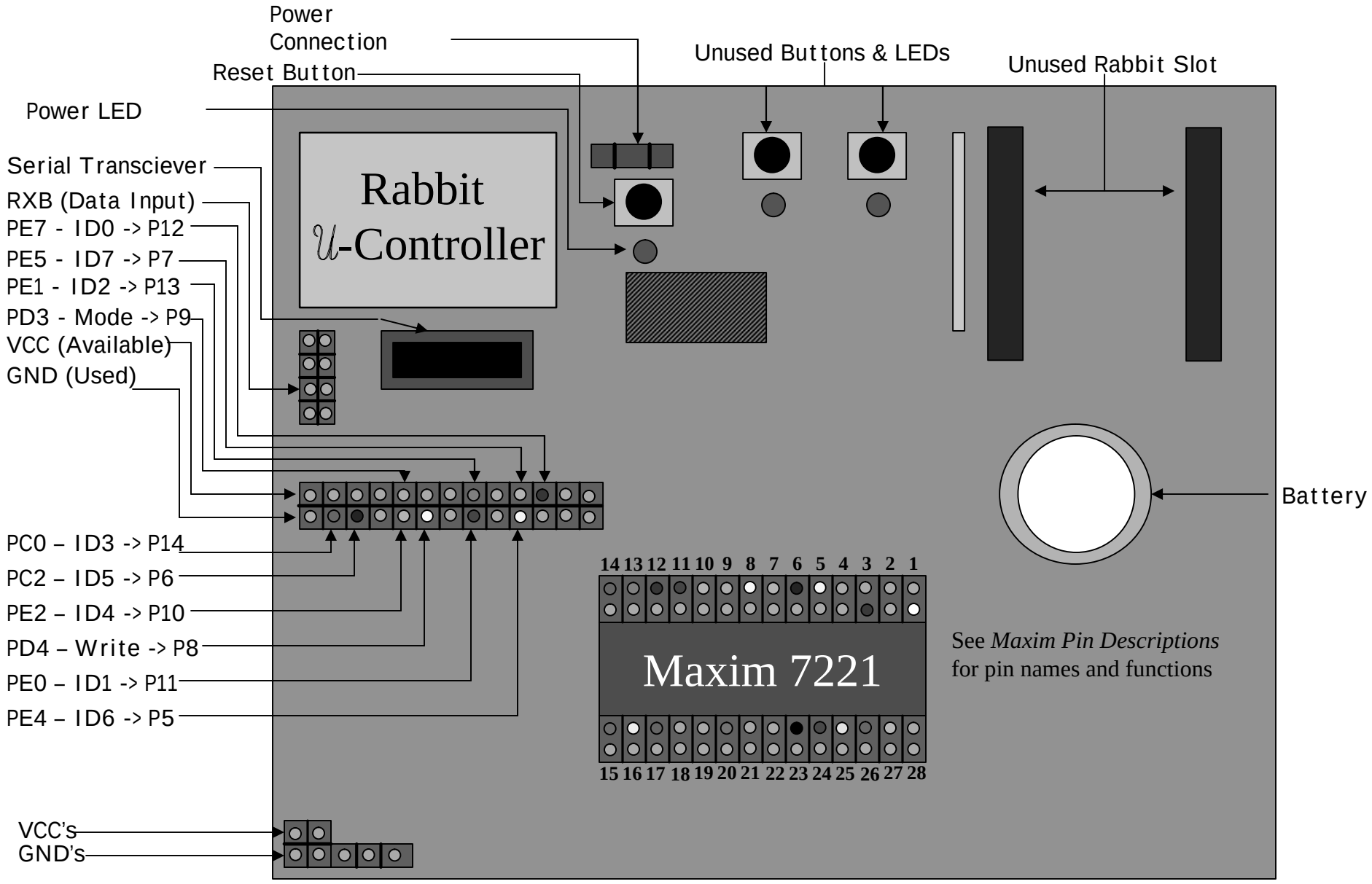
- 1 ms drop time
- 0.09 ms to drop the voltage to 17 volts




1K Resistor

- 0.2 ms drop time
- 0.03 ms to drop the voltage to 17 volts




Power Connection
Reset Button
Power LED
Serial Transciever
RXB (Data Input)
PE7 - ID0 -> P12
PE5 - ID7 -> P7
PE1 - ID2 -> P13
PD3 - Mode -> P9
VCC (Available)
GND (Used)
PC0 – ID3 -> P14
PC2 – ID5 -> P6
PE2 – ID4 -> P10
PD4 – Write -> P8
PE0 – ID1 -> P11
PE4 – ID6 -> P5
VCC's
GND's
Rabbit
U-Controller
Unused Buttons & LEDs
Unused Rabbit Slot
Battery
14 13 12 11 10 9 8 7 6 5 4 3 2 1
Maxim 7221
15 16 17 18 19 20 21 22 23 24 25 26 27 28
See *Maxim Pin Descriptions* for pin names and functions

Maxim Pin Descriptions

Pin 1: Segment C
Pin 2: Segment E
Pin 3: Segment B
Pin 4: Unused
Pin 5: ID6 – Connects to PE4 from Rabbit
Pin 6: ID5 – Connects to PC2 from Rabbit
Pin 7: ID7 – Connects to PE5 from Rabbit
Pin 8: Write – Connects to PD4 from Rabbit
Pin 9: Mode – Connects to PD3 from Rabbit
Pin 10: ID4 – Connects to PE2 from Rabbit
Pin 11: ID1 – Connects to PE0 from Rabbit
Pin 12: ID0 – Connects to PE7 from Rabbit
Pin 13: ID2 – Connects to PE1 from Rabbit
Pin 14: ID3 – Connects to PC0 from Rabbit
Pin 15: Digit 1 Power
Pin 16: Digit 2 Power
Pin 17: Digit 5 Power
Pin 18: Digit 8 Power (Unused)
Pin 19: VCC (+5 V input)
Pin 20: Digit 4 Power
Pin 21: Digit 7 Power (Unused)
Pin 22: Digit 6 Power (Unused)
Pin 23: Digit 3 Power
Pin 24: Segment F
Pin 25: Segment D
Pin 26: Segment G
Pin 27: Segment A
Pin 28: Ground

# Sx5 – Scope, Spending, Schedule, Spec's, & sFeatures

- Scope

Small change in the scope:

Instead of displaying leading 0's…

Display a blank:

The Maxim 7221A will display a blank on the specified digit when the value 0xFF is sent.

# Sx5 – Scope, Spending, Schedule, Spec's & sFeatures

## - Spending

Total: $[illegible]

-Display Case:

| | | |
|---|---|---|
| 15 – PMOS transistors @ $0.68 ea. | | ~ $10.20 |
| 20 – NMOS transistors @ $0.25 ea. | | ~ $5.00 |
| 1 – special shipping charge | | ~ $48.48 |
| 100 – Cable Ends (female) @ $0.08 ea | | ~ $8.00 |
| 100 – Cable Ends (male) @ $0.06 ea | | ~ $6.00 |
| Misc. Case Supplies (tape, fasteners, etc.) | | ~ $20.00 |
| Case Total: | | $[illegible] |

-Display System:

| | | |
|---|---|---|
| 1 – Rabbit Microcontroller | - $199.00 | ~ $199.00 |
| 5 – 5" 7-Segment Displays | - $12.95 | ~ $64.75 |
| Display Total: | | $[illegible] |

# Sx5 – Scope, Spending, Schedule, Spec's & sFeatures

- Schedule

| Milestones | Sept. '04 | | | | Oct. '04 | | | | Nov. '04 | | | | Dec. '04 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | W1 | W2 | W3 | W4 | W1 | W2 | W3 | W4 | W1 | W2 | W3 | W4 | W1 | W2 | W3 | W4 |
| Software Design | **Nate & Choi** | | | | | | | | | | | | | | | |
| Rabbit Serial Comm. | | **Nate** | | | | | | | | | | | | | | |
| Code Conversion | | | **Nate & Choi** | | | | | | | | | | | | | |
| Maxim Operation | | | | | **Choi** | | | | | | | | | | | |
| Power Supply System | | | | **Karl** | | | | | | | | | | | | |
| Part Procurement | | | **Karl** | | | | | | | | | | | | | |
| Part Integration | | | | | | | **ALL** | | | | | | | | | |
| Individual Component Testing | | | | | | **ALL** | | | | | | | | | | |
| Integrated System Testing | | | | | | | | | | | **ALL** | | | | | |
| Interception & Extraction | | | | | | | Nate | | | | | | | | | |
| Maxim Operating | | | | | | | | | Choi | | | | | | | |
| Large Displays Working | | | | | | | | | | | | | | | Karl | |

# Sx5 – Scope, Spending, Schedule, Spec's & sFeatures

- Spec's
  - -RS232 Input via ¼" phono plug
    - 9600 Baud
    - Even Parity
    - 8 data bits
    - 1 stop bit
  - -Standard 120V AC input
  - -Rabbit RCM2200
  - -Maxim 7221A display driver
  - -5 – 5" 7-segment displays
  - -120V AC -> 24V DC power supply

- sFeatures
  - -Continuous Display of Current Event & Heat
  - -Compatibility with Colorado Timing System's System 5 Timing Computer
  - -Long Range Visibility

Special Thanks to Prof. Frenzel & Prof. Wall for making it viewable from more than 2 feet!

Questions?