MICROMOUSE

Final Design Review

for **EE 495C Capstone Senior Design**

Team 4: **B**ill **D**ixon, **E**ric **R**ico, **P**atrick **D**owell, **S**onja **A**bbey

Faculty Advisors: **M**aarten **U**ijt de **H**aag, **W**outer **P**elgrum

June 3, 2010

I. **Introduction**
II. Technical Design Summary
III. Test Procedures and Results
IV. Program Management Summary
V. Conclusions

Original Goals

- Win 2010 Micromouse competition
- Improve upon common Micromouse design
  - Speed
  - Sensing Range
  - Reliability
- Construct full-scale test maze
- Document design for good handoff
  - Prepare new team for next year's work

Requirements - Performance

- Open-loop sprint speed capability ≥ 2 m/s
- Travel speed while mapping ≥ 0.1 m/s
- Sense walls at least 1 m away
  - Mix of short- & long-range sensors is desirable
- Total running time without battery change > 30 minutes
  - Mapping runs will consume majority of this time

I. Introduction
II. **Technical Design Summary**
III. Test Procedures and Results
IV. Program Management Summary
V. Conclusions

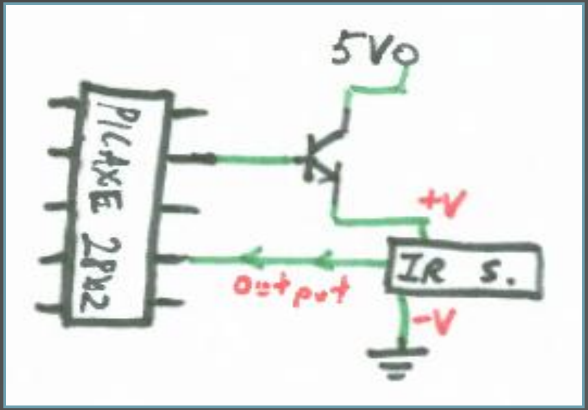
Final Trade Studies
Refresher

- Recall from previous design reviews:
  - PICOne ("pick one") baseline robot chosen over AIRAT II
  - Li-Ion batteries chosen for long run time
    - Original PICOne calls for 9V alkaline

Final Trade Studies

Long-Range Sensing

- Original performance requirements mention long-range wall-sensing capability
- A sensor was tested and ready to be installed mid-spring

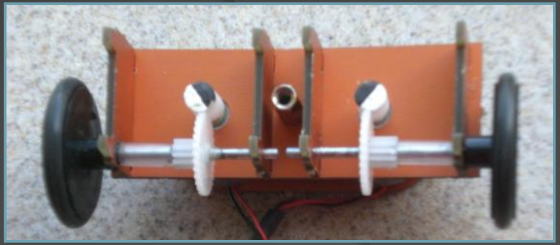


- However…

Final Trade Studies
Long-Range Sensing

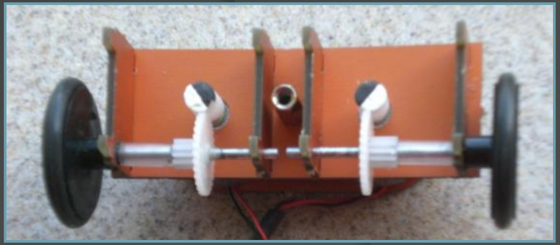
- Many upgrade possibilities were at hand
  - Usefulness of long-range sensing was re-evaluated
- Team decided not to install long-range sensors due to limited utility
  - This allowed time for more useful upgrades

Final Trade Studies
Speed Increase

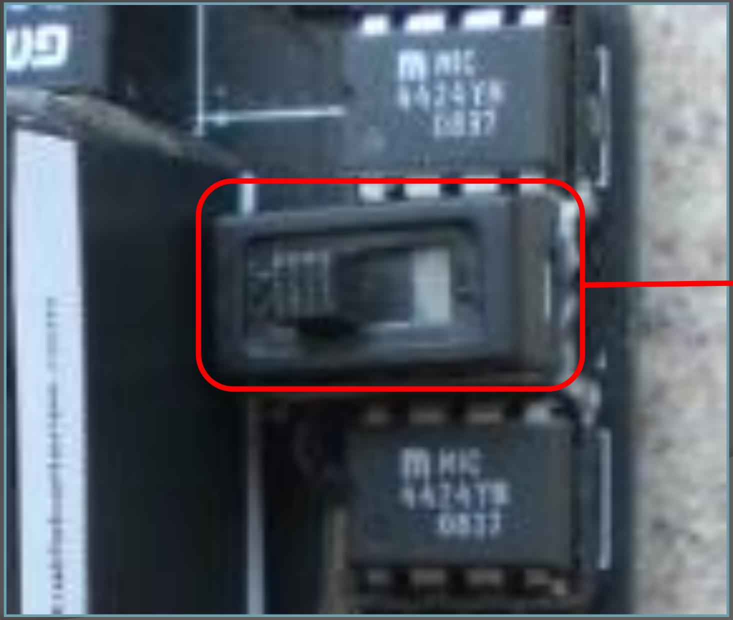


- Performance requirement – **2 m/s** sprint speed
  - Very fast
  - During spring, team found that mouse would need complete re-design for this speed
- *Should the team attempt to meet the requirement*?
- An attempt was made to achieve it
  - This could be done either by using different motors or increasing V_{SUPPLY} to the driver IC

Final Trade Studies
Speed Increase

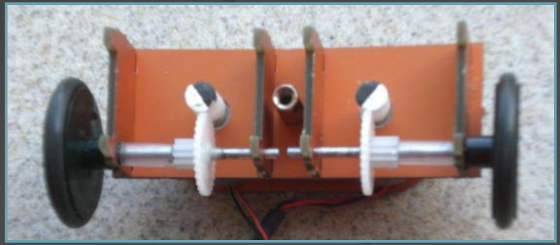


- Drive voltage of motors was increased
- Switch allows user to select "safe" or "fast" mode
  - Safe = ~0.3 m/s*　　Fast = ~0.8 m/s*



*These speeds depend on how often the mouse must stop and "think"

Final Trade Studies
Speed Increase

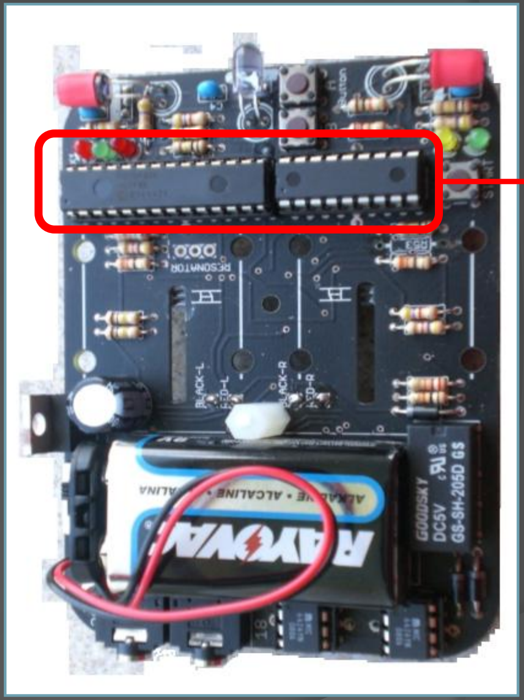


- Further increase in speed requires three things:
  - Different motors (brushless?)
  - Stronger driver IC
  - Better traction management

Design Summary

Hardware Highlights – Processor

- PICOne kit uses PICAXE 28X1 and 18X
  - Rather slow, 8 MHz



- What is a PICAXE?
  - PICmicro with bootstrap code
  - Can be programmed in BASIC
- PICAXE is at the heart of our micromouse

Design Summary
Hardware Highlights – Processor

- Processors were replaced with single chip – 28X2
  - Also added 10 MHz resonator
    - Allows 40 MHz clock
  - Much faster – 40 MHz vs. 8 MHz

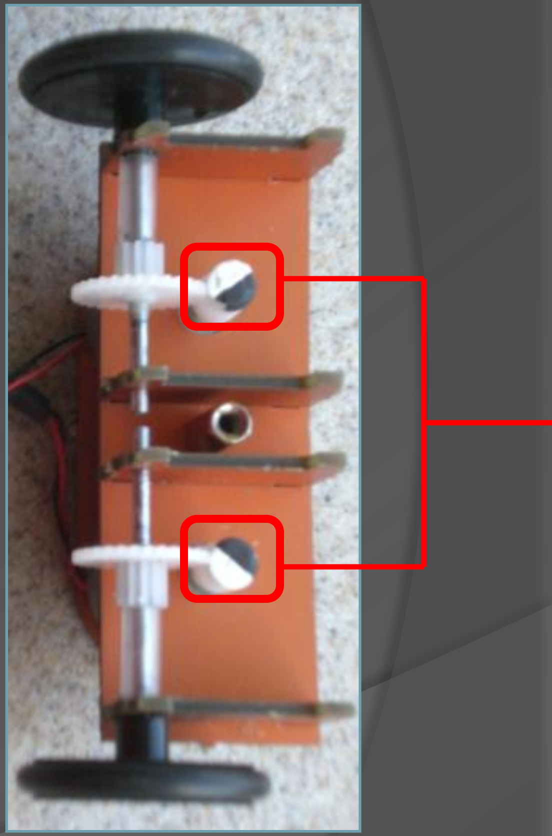
- This enables:
  - Faster "thinking"
  - More program and data memory
  - All configurable I/O
  - More board space for other components

Design Summary
Hardware Highlights – Drive System

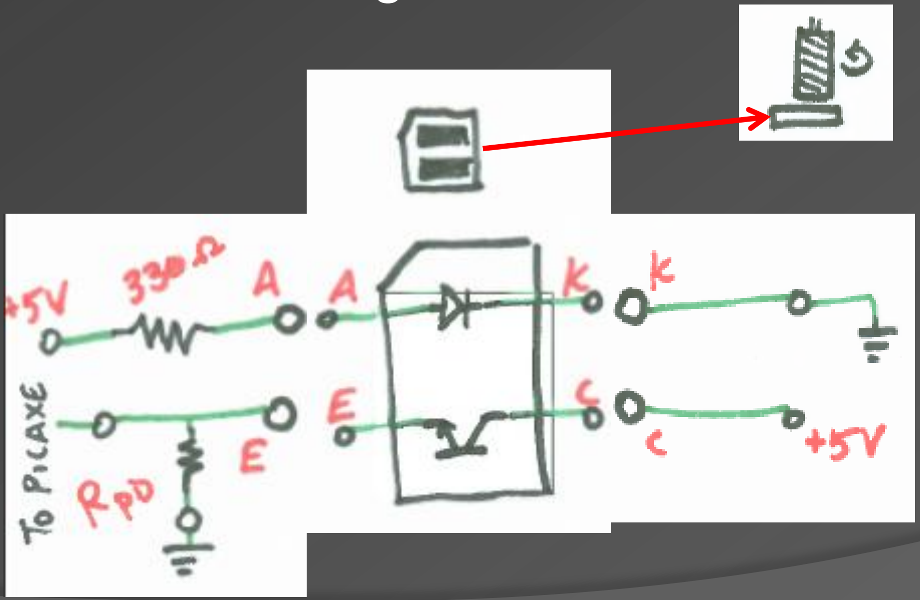
- Drive system centered on 2 brushed DC motors
  - Commonly used in CD players
- Motors driven by 2-channel MOSFET driver chip
- Driver IC controlled by PWM from PICAXE
- Right motor is reversible via relay

Design Summary
Hardware Highlights – Drive System

- Control loop closed by 2 encoders
  - Reflector on L and R drive shafts
  - Output of these is used in software



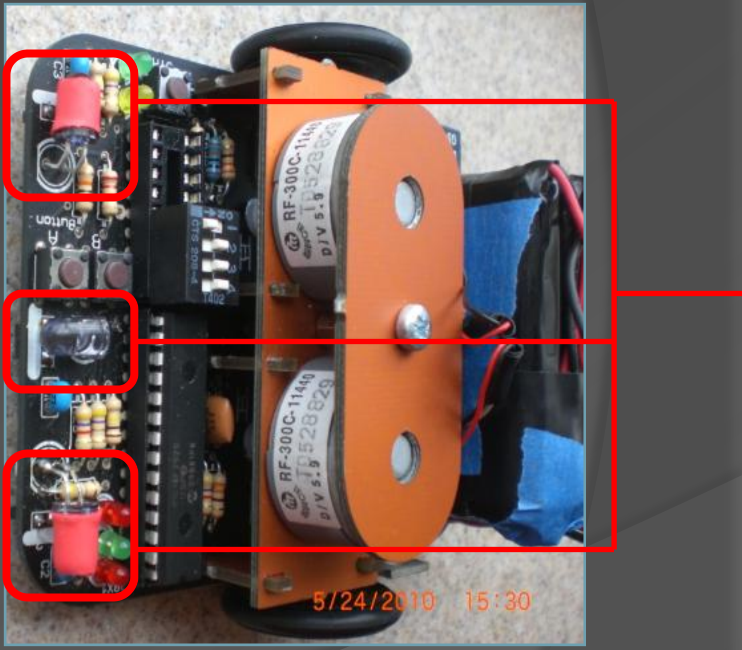
- Encoder design:



Design Summary

Hardware Highlights – Wall Sensors

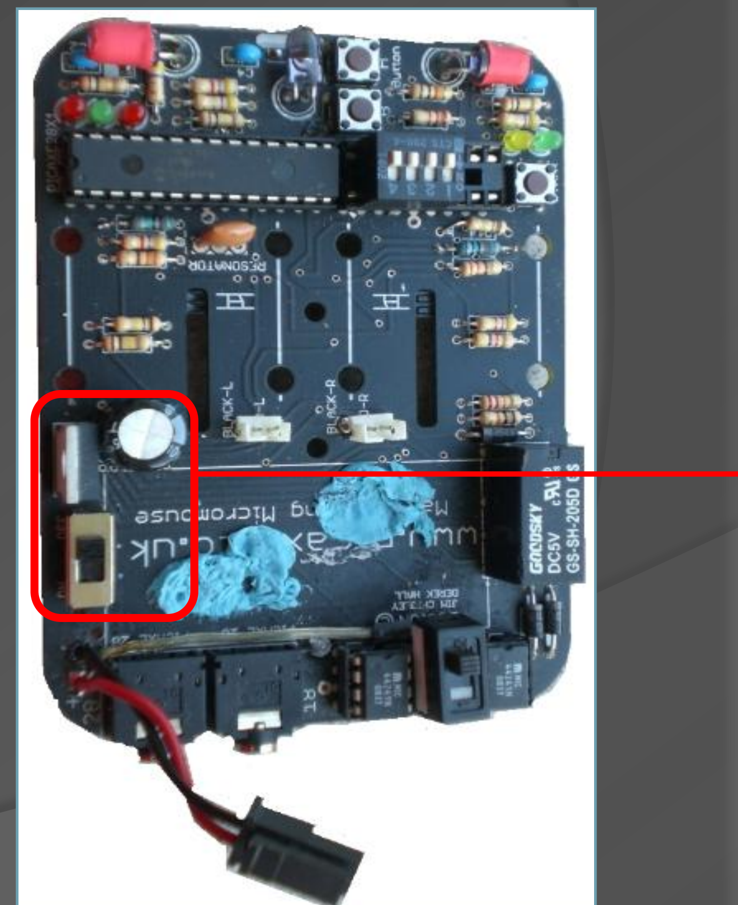
- 3 IR LED "headlights"
- Magnitude of returning light read by PICAXE ADC pins
  - Used in software
  - Treated as wall distance
- AC coupled to PICAXE for ambient light rejection



Design Summary

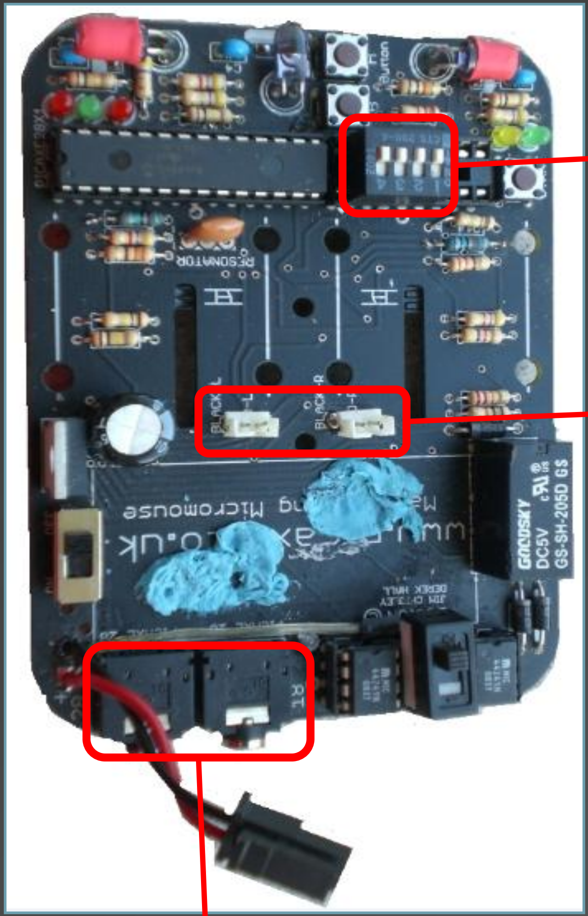
Hardware Highlights – Power

- 2-cell Li-ion battery pack
  - 7.4V, 1800 mAh
  - Run time > 4 hours
- Board uses 5V regulated supply
- Solderless connectors help achieve a modular design



Design Summary

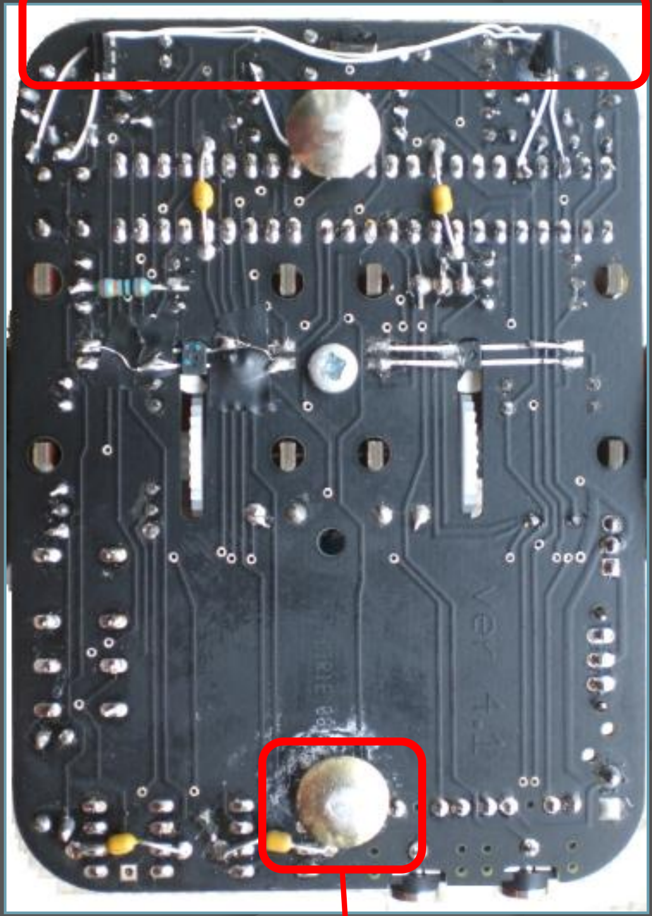
Hardware Highlights – Other



Expansion DIP Switches

JST Motor Connectors

Serial COM Jacks



PICAXE 28X2 Re-Wire

Thumbtack Skids

Design Summary
Hardware Highlights – Other

Mouse Costume

Full-Size Test Maze

Design Summary

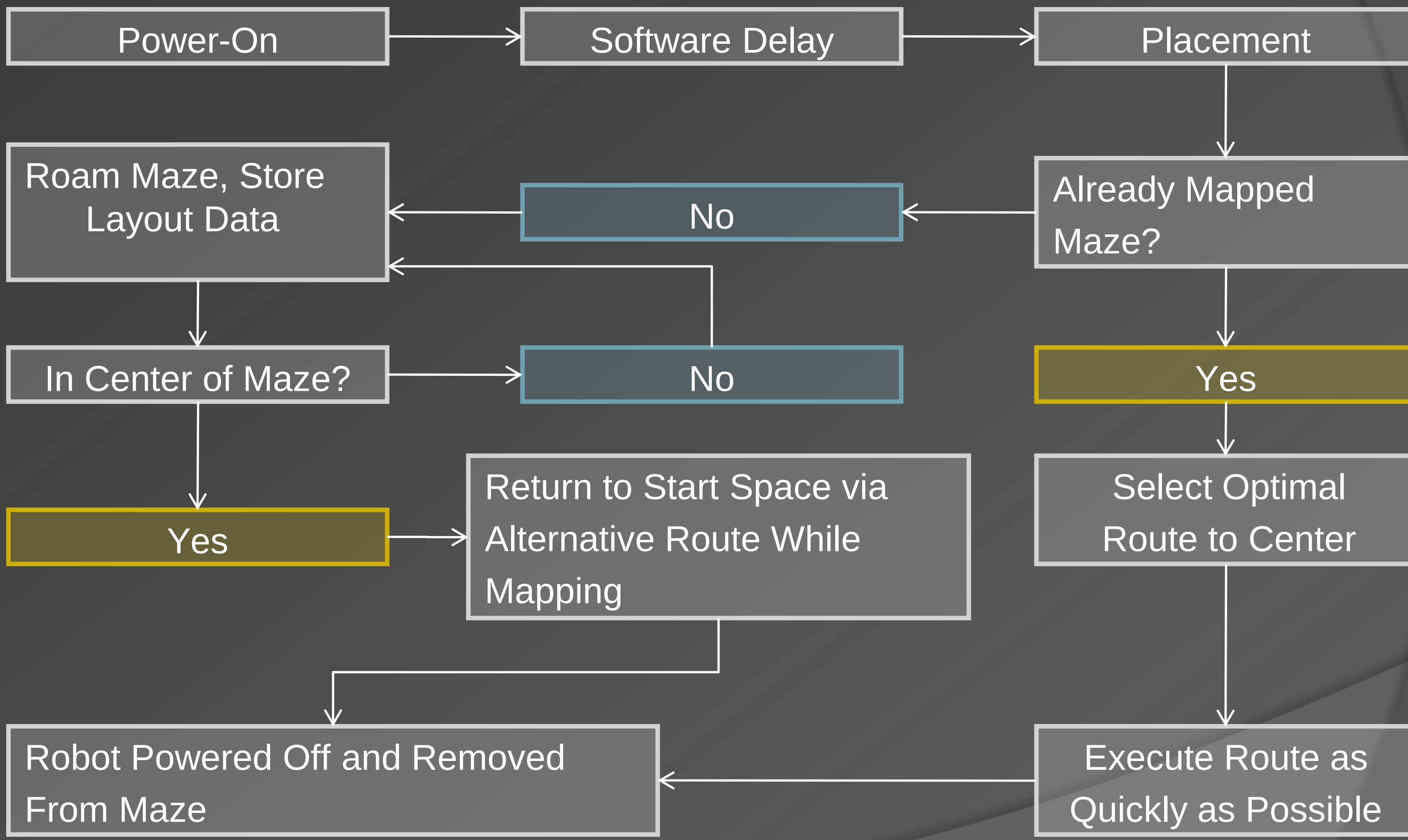
- “-ilities” to be addressed later

Design Summary
Software

- PICAXE-28x2
- Conversion from 28x1 and 18x1 to 28x2
  - New program for 28x2 and higher PICAXEs
  - Set frequency of higher clock speed in programming
  - Provided calibration settings from 28x1 and 18x1
  - Modified motor control loops
  - Maze Solving Algorithm

Design Summary

Functional Flow - Software



Design Summary

Software

- PICone
  - Programming in BASIC
  - Implemented two processors into one

```
symbol maze_start=$F0          'Position in maze map after a reset (RAM)
symbol maze_center=$87         '$87 $87 is centre for 16*16 maze (RAM)



'Inits from 28X1

main:

symbol left_motor=1000         'sets left motor speed for straight line max 1023
symbol right_motor=988         'sets right motor speed for straight line max 1023
symbol left_wall=8             'sets left wall detection
symbol right_wall=8            'sets right wall detection
symbol front_wall=5            'sets front wall detection at 12cm
symbol reset_frontwall=12      'resets wheel counter at 8.5cm if deadend is found
symbol left_straighten=25      'value of left sensor if no straightening required
symbol right_straighten=25     'value of right sensor if no straightening required
symbol straight_before_right=10 'sets distance of short straight before right turn
symbol angle_right=82          'sets amount of right turn
symbol straight_after_right=32 'sets distance of short straight after right turn (lower number to travel further)
symbol straight_before_left=10 'sets distance of short straight before left turn
symbol angle_left=83           'sets amount of left turn
symbol straight_after_left=41  'sets distance of short straight after left turn (lower number to travel further)
symbol straight_turnround=40   'distance travelled into deadend before turn round
symbol angle_turnround=84      'sets amount of turn round
```

Design Summary
Software

- PICone
  - Starting loops
  - A button and start button

```
'Setup for new run

if maze_reset_button=0 then
       gosub clear_maze  'if "A" button pressed then reset the maze map
else
       gosub clear_maze_bits   'Don't erase wall data, just restart maze
endif

'Ported from 18X... Start of run

switch on yellow_led

let target=maze_center  'first run to the center
let pos=maze_start-16   'start in bottom left square (mouse starts with forward move so assume start one square forward)
let direc=0             'start facing north
let solvit=0

gosub solve_maze        'solve the maze
switch off yellow_led   'mouse ready to go when yellow led is off
```

```
'Now start the control routine (derived from 28X1 code)

             low C.5          'IR leds off
label_D:     pwmout C.1,255,0 'stop right motor
label_14:    pwmout C.2,255,0 'stop left motor
label_1B:   let b45= 20       'mouse start square requires added wheel counts to straight line count
label_22:   high B.6          'middle green led on
label_29:   if maze_reset_button=0 then gosub list_maze    'if yes output maze to PC
            if pinC.6=1 then label_29 'wait for start button
            low B.6           'middle green led off
            pause 2000
```

Design Summary
Software

- PICone
  - Read left and right walls, Red light turns ON
  - Read front wall, Green light turns ON
  - Wall Straightening Loops

```
label_105:  if b45> 61 then label_131   'straight line wheel counter end of current move
            if b40< left_wall then label_126  'straighten on left wall if available
            if b40> left_straighten then label_153
label_14C:   pwmout C.2,255,850    'slower speed for left motor to straighten mouse
             pwmout C.1,255,right_motor
            goto label_49
```

```
label_126:  if b42< right_wall then label_3B  'straighten on right wall if available
            if b42< right_straighten then label_153
            goto label_14C
```

```
label_153:   pwmout C.1,255,850    'slower speed for right motor to straighten mouse
             pwmout C.2,255,left_motor
            goto label_49
```

Design Summary
Software

- PICone
  - Maze Solving Algorithm
    - Select what to do in situations

```
select case sensor
	case 3
		let b8=1			'if walls both sides and no front wall always go forward
	case 5
		let b8=2			'If walls to front and left always go right
	case 6
		let b8=3			'If walls to front and right always go left
	case 7
		let b8=0			'If in dead end always turn around
	else
		let	b8=map_walls & %00000011 'Else use the maze map to decide where to go
		select case direc 'convert to wall map bits depending on the direction of the mouse
			case 0
				lookup b8,(1,2,0,3),b8 'mouse facing north
			case 1
				lookup b8,(3,1,2,0),b8 'mouse facing east
			case 2
				lookup b8,(0,3,1,2),b8  'mouse facing south
			case 3
				lookup b8,(2,0,3,1),b8 'mouse facing west
		endselect
endselect
```

Design Summary

Software

- PICone
  - Determine if have been in block
  - Matrix to determine position in maze

```
select case direc ;convert to wall map bits depending on the direction of the mouse
case 0
      lookup sensor,(0,16,64,80,32,48,96,112),b3 'mouse facing north
case 1
      lookup sensor,(0,32,128,160,64,96,192,224),b3 'mouse facing east
case 2
      lookup sensor,(0,64,16,80,128,192,144,208),b3   'mouse facing south
case 3
      lookup sensor,(0,128,32,160,16,144,48,176),b3 'mouse facing west
endselect

let map_walls = map_walls & %00001111
let map_walls = map_walls | b3
let w_visited = 1                 'set visited bit
write pos,map_walls               'store maze walls in map
```

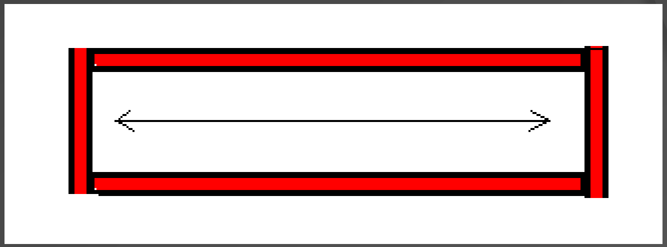
```
'Blank maze data
data $00, ($30,  $20, $20, $20, $20, $20, $20, $20, $20, $20, $20, $20, $20, $20, $20,  $60)
data $10, ($10,  $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00,  $40)
data $20, ($10,  $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00,  $40)
data $30, ($10,  $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00,  $40)
data $40, ($10,  $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00,  $40)
data $50, ($10,  $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00,  $40)
data $60, ($10,  $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00,  $40)
data $70, ($10,  $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00,  $40)
data $80, ($10,  $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00,  $40)
data $90, ($10,  $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00,  $40)
data $A0, ($10,  $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00,  $40)
data $B0, ($10,  $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00,  $40)
data $C0, ($10,  $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00,  $40)
data $D0, ($10,  $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00,  $40)
data $E0, ($10,  $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00, $00,  $40)
data $F0, ($D4,  $90, $80, $80, $80, $80, $80, $80, $80, $80, $80, $80, $80, $80, $80,  $C0)
```

I. Introduction
II. Technical Design Summary
III. **Test Procedures and Results**
IV. Program Management Summary
V. Conclusions

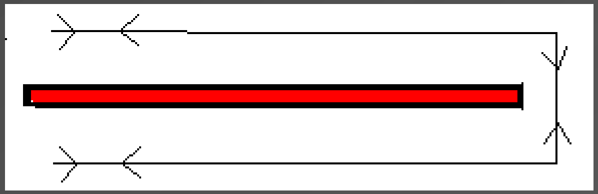
Test Procedures and Results

Test Procedures Executed

- Straight Line Test



- Turning off wall test

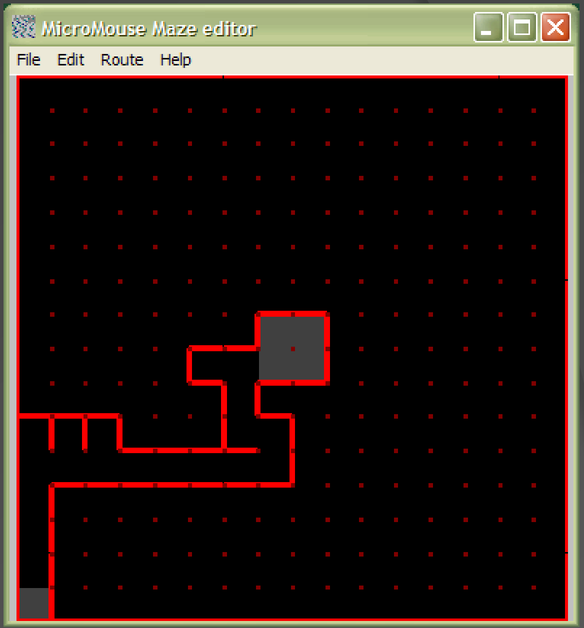


- Speed Tests
  - Mapping
  - Pre-Mapped Run
- Long-Range Sensor

Test Procedures and Results

Various Tests

- MicroMouse Maze Editor



- Debug Program

```
'BASIC converted from flowchart
'Test 3 debug 28X1.CAD


main:
                setfreq m8
                pwmout 1,255,0
                pwmout 2,255,0
label_37:      debug  b1
                high portc 5
               readadc  0,b0
               readadc  1,b1
               readadc  2,b2
                low portc 5
               pause 20
               goto label_37
```

Test Procedures and Results

Process in Running Robot

1. User Input

User sets the robot in starting position and powers it on.

2. Mapping

The robot stores the maze-layout data as it explores the maze.

3. Route Selection

Upon completion of the mapping run, the robot selects the optimal route to the maze's center.

4. Route Execution

The robot travels the selected optimal route. During this run the robot travels faster than in the mapping run.

5. Exit Maze

User removes the Micromouse from the maze by hand.

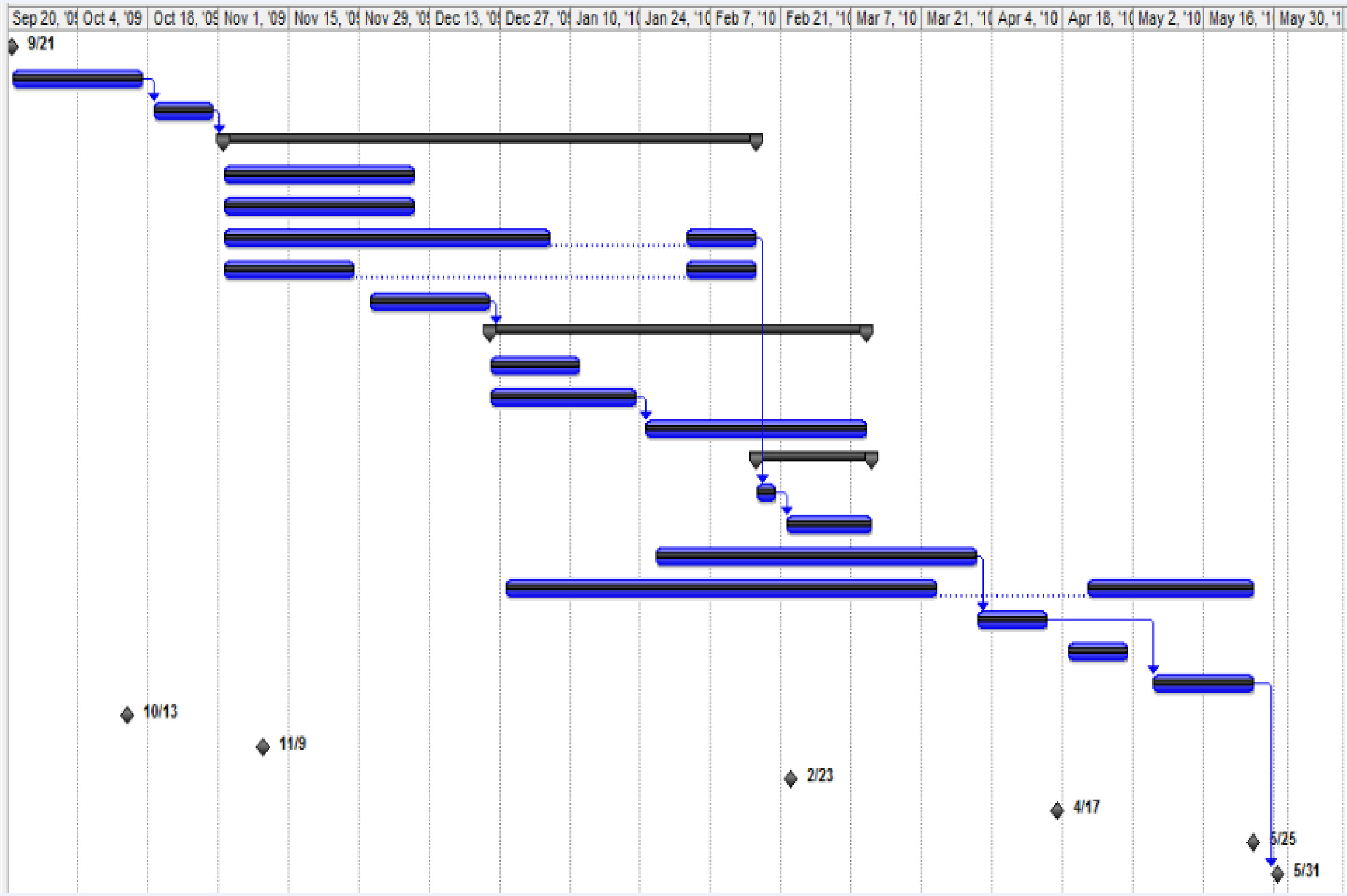
Requirements Compliance

	Requirement	How it's met by design
1	2.0 m/s sprint speed	PICone: Overdriven motors REQUIREMENT NOT MET
2	0.1 m/s maze-mapping speed	PICone: Drive system achieves this speed easily REQUIREMENT MET
3	1.0 m wall-sensing range	PICone: Integrates extra long-range IR sensor DID NOT IMPLEMENT LONG-RANGE
4	Intelligent maze-solving	PICone: Stock software performs this function REQUIREMENT MET
5	30-minute run time	PICone: Stack high-capacity Li-Ion batteries on unit REQUIREMENT MET
6	Win Competition	PICone: Capable of winning the competition REQUIREMENT MET

I. Introduction
II. Technical Design Summary
III. Test Procedures and Results
IV. **Program Management Summary**
V. Conclusions

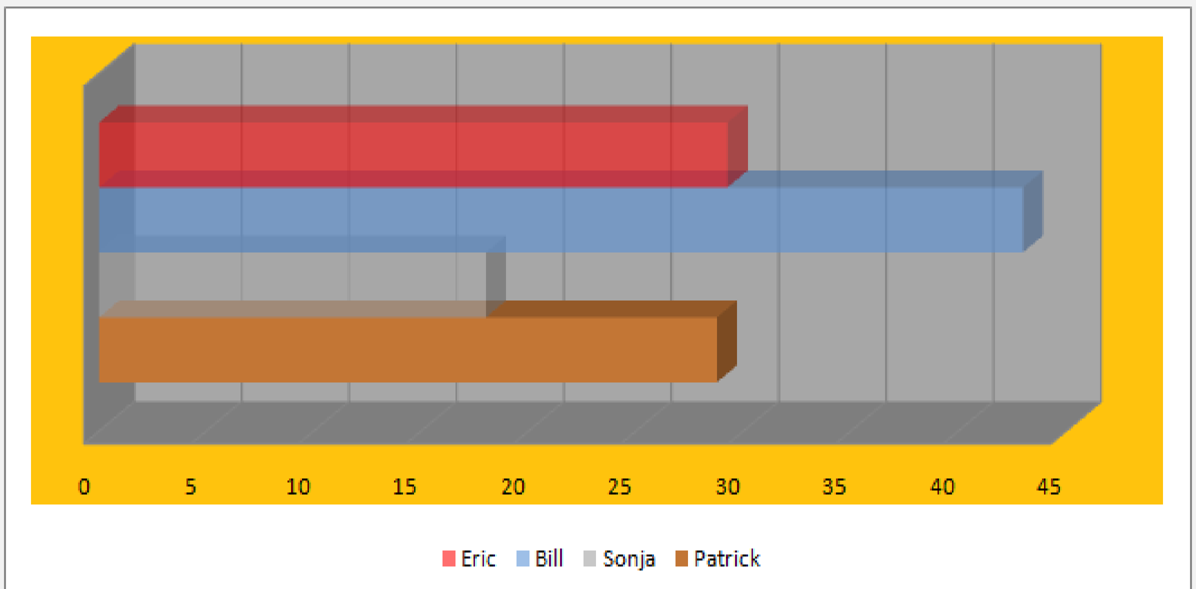
Schedule – Gantt Chart

	Task Name	Duration
1	Project Kickoff	0 days
2	Conceptual Brainstorming	20 days
3	System Requirements Creation	10 days
4	Trade Studies	76 days
5	Component Acquisition Method	28 days
6	Processor Family	28 days
7	Sensing Technology	57 days
8	Power Supply (Battery Type)	30 days
9	Non-Sensor Hardware Acquisition	18 days
10	Non-Sensor Hardware Assembly	54 days
11	Programmable Mock-Up	12 days
12	Working Prototype	21 days
13	Evaluation, Initial Testing	33 days
14	Sensors	18 days
15	Selection	4 days
16	Acquisition	14 days
17	Software Assembly and Adaptation	47 days
18	Maze Construction	86 days
19	Final Verification, Testing, Debugging	12 days
20	Processor Replacement	10 days?
21	Documentation Writing	14 days
22	System Requirements Review	0 days
23	Preliminary Design Review	0 days
24	Critical Design Review	0 days
25	IEEE Micromouse Competition	0 days
26	Project Complete	0 days
27	Final Design Review	0 days



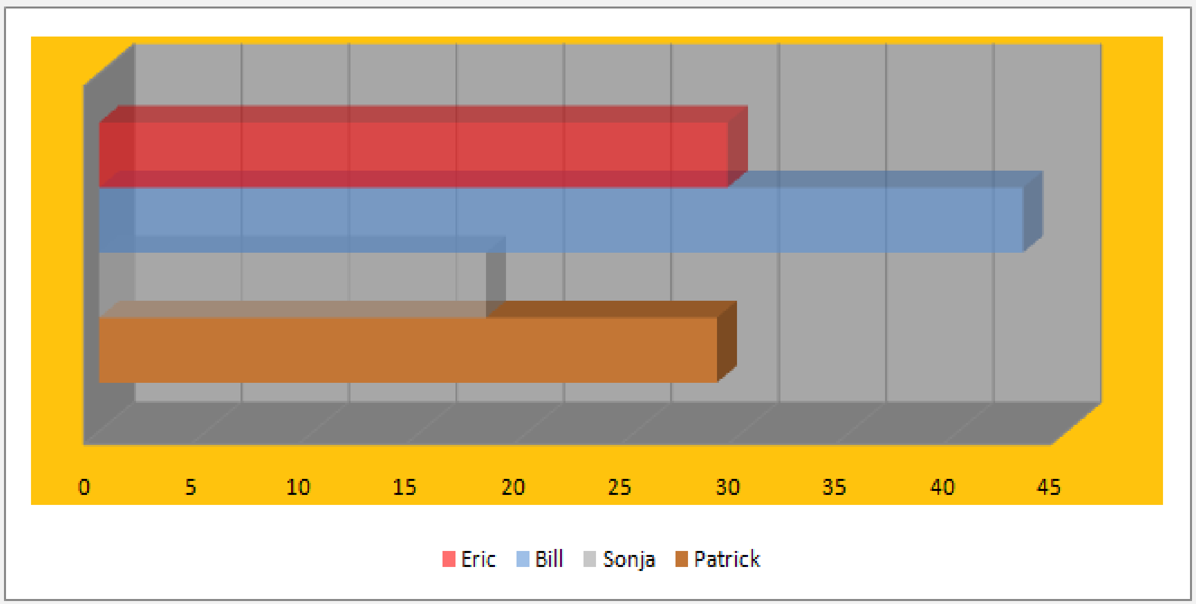
Schedule – Time Sheet

Personnel																					Totals
Patrick Dowell	2	0.75	1	0.5	1	1	1	3	2	3	2	3	5		2	1.5					28.75
Sonja Abbey	2	2	2	4	2	2	2	2													18
Bill Dixon	1	1	1	1	1	4	1	2	7	1	1	1	3	2	4	3	5	2	2		43
Eric Rico	1	2	1	0.75	0.5	2	3	5	3	11											29.25



Schedule – Time Sheet

Personnel																					Totals
Patrick Dowell	2	0.75	1	0.5	1	1	1	3	2	3	2	3	5		2	1.5					28.75
Sonja Abbey	2	2	2	4	2	2	2	2													18
Bill Dixon	1	1	1	1	1	4	1	2	7	1	1	1	3	2	4	3	5	2	2		43
Eric Rico	1	2	1	0.75	0.5	2	3	5	3	11											29.25



Schedule – Time Sheet

	BILL DIXON	
Date	**Description**	**Time Spent**
2-Mar-10	Meeting with group	1
2-Mar-10	Attempted to charge one of the AiratII battery packs	1
3-Mar-10	Used new smart charger for Airat II batteries. Kicked off and re glued some misaligned Maze posts.	1
4-Mar-10	Charged Airat II with Sonja and Patrick	1
4-Mar-10	Progress Report with group	1
4-Mar-10	Made a wall-following program with sensor modulation (cancels ambient IR) for the PICone and ser	4
17-Mar-10	Senior Design Research PICOne motors (They're actually CD/DVD drive motors) and DC motors in	1
19-Mar-10	repairing PICOne with one of the new sensors	2
28-Mar-10	debugging PICone encoder, installing new photomicrosensor, changing encoder pull-down resistor and pai	7
30-Mar-10	getting new Li-Ion batteries into packs and onboard PICone	1
2-Apr-10	testing PICone in maze	1
6-Apr-10	soldering battery pack together and getting Kadi in Avionics to put connectors on PICone board and Li-Ion	1
7-Apr-10	making a batch of new maze posts that sit correctly on a flat surface and match the wall height better (stil	3
8-Apr-10	installing switch on PICone (main) that allows motors to be driven by either regulated 5V or battery voltage	2
11-Apr-10	putting together backup PICone and messing with right encoder pulldown resistor to get it to work.	4
13-Apr-10	installing pins on PICone board and connectors on motors (both main and backup) so that it can be disas	3
15-Apr-10	repairing IR LED breakage and helping Rico and Patrick fine-tune both PICones (main and backup)	5
20-Apr-10	doing 18X -> 28X2 pin assignments, data gathering, & parts selection (resonator & some small electronics	2
21-Apr-10	building frame for 2nd half of maze.	2

Resource Allocation

Class Funds			**$500.00**
Maze Materials			($400.00)
Sensors			($45.00)
Batteries			($55.00)
Remaining			$00.00

Resource Allocation

Customer Funds			$1,331.00
AIRAT II			($734.00)
PICone (x2)			($542.00)
Resonator, PICAXE, etc			($35.00)
Backup Motors			($20.00)
Remaining			$0.00

Resource Allocation

- Personnel Hours
  - The team logs approximately 10 man-hours per week

-ilities

- Reliability
  - Numerous test runs in various maze layouts proved the reliability of the design before competition
- Flexibility
  - Replacing the two processors with a single processor increased the number of I/O pins for more sensors and opened up room on the board
- Maintainability
  - Multiple battery packs reduced downtime in case of battery death
  - Purchase of extra parts makes replacement of damaged equipment quick and simple

Meeting Statistics

Date	Present	Planned	Minutes	Duration	Effective
Winter					
01/12/10	B, E, P, S, W, M	Planned	No	45 min	Yes
01/19/10	B, E, P, S, W, M	Planned	No	1 hour	Yes
01/26/10	B, E, S, W, M	Planned	Yes	1 hour	Yes
02/02/10	B, E, P, S, W, M	Planned	Yes	1 hour	Yes
02/09/10	B, E, S, W, M	Planned	Yes	1 hour	Yes
02/16/10	P, S, W, M	Planned	Yes	1 hour	Yes
02/23/10	B, E, P, S, W, M	Planned	No	1 hour	Yes
03/02/10	B, E, P, S, W, M	Planned	Yes	1 hour	Yes
03/09/10	B, E, P, S, W, M	Planned	No	1 hour	Yes
Spring					
04/06/10	B, E, S, W	Planned	Yes	1 hour	Yes
04/13/10	B, E, P, S, W, M	Planned	No	1 hour	Yes
04/20/10	B, P, S, W, M	Planned	Yes	1 hour	Yes
04/27/10	B, E, S, W, M	Planned	No	1 hour	Yes
05/04/10	B, E, P, S, W, M	Planned	No	45 min	Yes
05/11/10	B, E, P, S, W, M	Planned	No	1 hour	Yes
05/19/10	B, E, M	Spontaneous	No	1 hour	Yes
05/25/10	B, E, P, S, W	Planned	No	45 min	Yes
06/02/10	B, E, P	Spontaneous	No	1 hour	Yes

*Members: Bill, Eric, Pat, Sonja, Wouter, Maarten

I. Introduction
II. Technical Design Summary
III. Test Procedures and Results
IV. Program Management Summary
V. **Conclusions**

Conclusion

- Project met expectations for performance at competition
  - Made it to the center of the maze in the first run and team placed first in competition
- Project met expectations for design
  - Robot successfully sensed maze walls and navigated to the center

Recommendations

- Software Modifications
  - Vary speed in software instead of with a switch
- Hardware Modifications
  - Rewire motor driver – only have supply voltage from battery
  - Replace wheels – increase traction, reduce slip
  - Odometer – won’t be messed up if slipping occurs

Recommendations

- Meeting Planning
  - Meeting minutes help
  - Be clear in tasks to accomplished and by whom before next meeting
- Competition
  - Prepare to get fastest run in first run

Questions?