The Future Soldier

Group 4
Vincent Leone
Elliot Levy Bencheton
Asim Chaudhry

Overview

- Objective
- Proposed Solution
- Introduction
- Hardware
- Software
- Cost Estimation
- Future Plans/Improvements
- Conclusion
- Acknowledgements

Objectives

- To increase strategic coordination for military personnel by increasing the situational awareness of infantrymen and their commanding officers
- To help coordinate emergency service efforts in the arena of complex urban environments

Proposed Solution

- To use a combination of GPS coordinates and human dead reckoning to find the exact position of a person whether indoors or outdoors
- To share encrypted positioning and wayfinding data between multiple individuals over Radio Frequency (RF)
- Display positioning and sensor data on an easy to use interface like Google Maps

Introduction

- Navigational System
  - Positioning
    - -GPS (primary)
      - -Requires at least 4 satellites, otherwise a dead reckoning system is used as a fallback
    - -Dead Reckoning (secondary)
      - -Compass
      - -Sonar/Switches
  - Way finding
    - -Preloaded waypoint definitions
    - -Heads up display LED navigation
    - -Piezo speaker proximity actuation

Introduction cont.

- Interface
  - -3 arm mounted push buttons for control
  - -2 foot mounted limit switches for step counting
  - -Sonar step length measurement
  - -Compass for angular rotation
  - -Global positioning system for latitude and longitude
  - -Temperature sensor for human body monitoring
- Display
  - Serial LCD for displaying coordinates
  - LED Heads up display for wayfinding

Introduction cont.

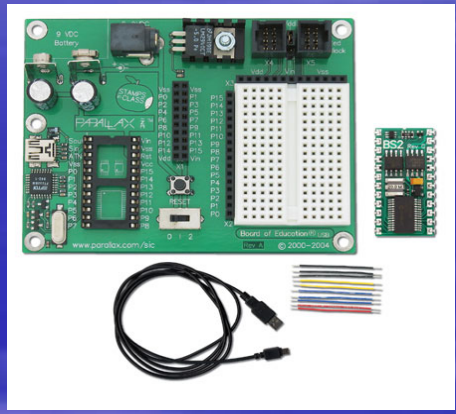
- Communication

  -RF Transmitter on user to send GPS, dead reckoning and distress signals

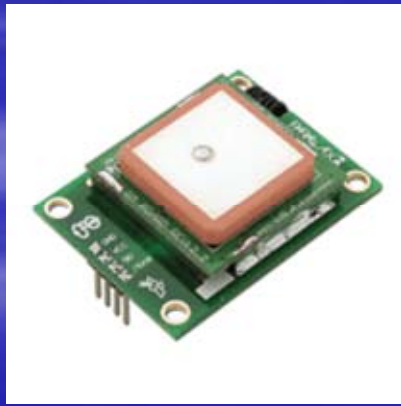
  -RF Receiver at base where user's position can be monitored in real time on "Google Maps"

Hardware

- Basic Stamp 2 with Board of Education
  - 16 Serial I/O pins
  - 2k EEPROM
  - 32 bytes of RAM



- GPS Module
  - Provides Current Latitude, Longitude, Altitude, Time, Compass Heading and Bearing, Speed, and Number of Satellites
  - Exact position of user outdoors



Hardware II

- Digital Compass – HM55B
  - Dual-Axis Magnetic Field Sensor
  - 6-bit(64 Direction) Resolution
  - Allows user to know direction of waypoint

- LED's for Heads Up Display
  - 2 LED's that points user to direction of waypoint.

- Piezo Speaker
  - Actuator placed in helmet for audio feedback at waypoint

Hardware III

- RF Transmitter/Receiver
  - 433.92 Mhz (UHF)
  - 500ft+ range based on environment
  - Allows communication between multiple Basic Stamps to send and receive position data

- Ultra Sonic Sonar
  - Distance measurement for step-length of user
  - Range: 2cm – 3m

- Limit Switches for Pedometer
  - Buttons placed on boot to count number of steps taken in Dead Reckoning mode

Hardware IV

- LM34 Temperature Sensor

  Used in conjunction with an ADC0831 Analog-Digital converter

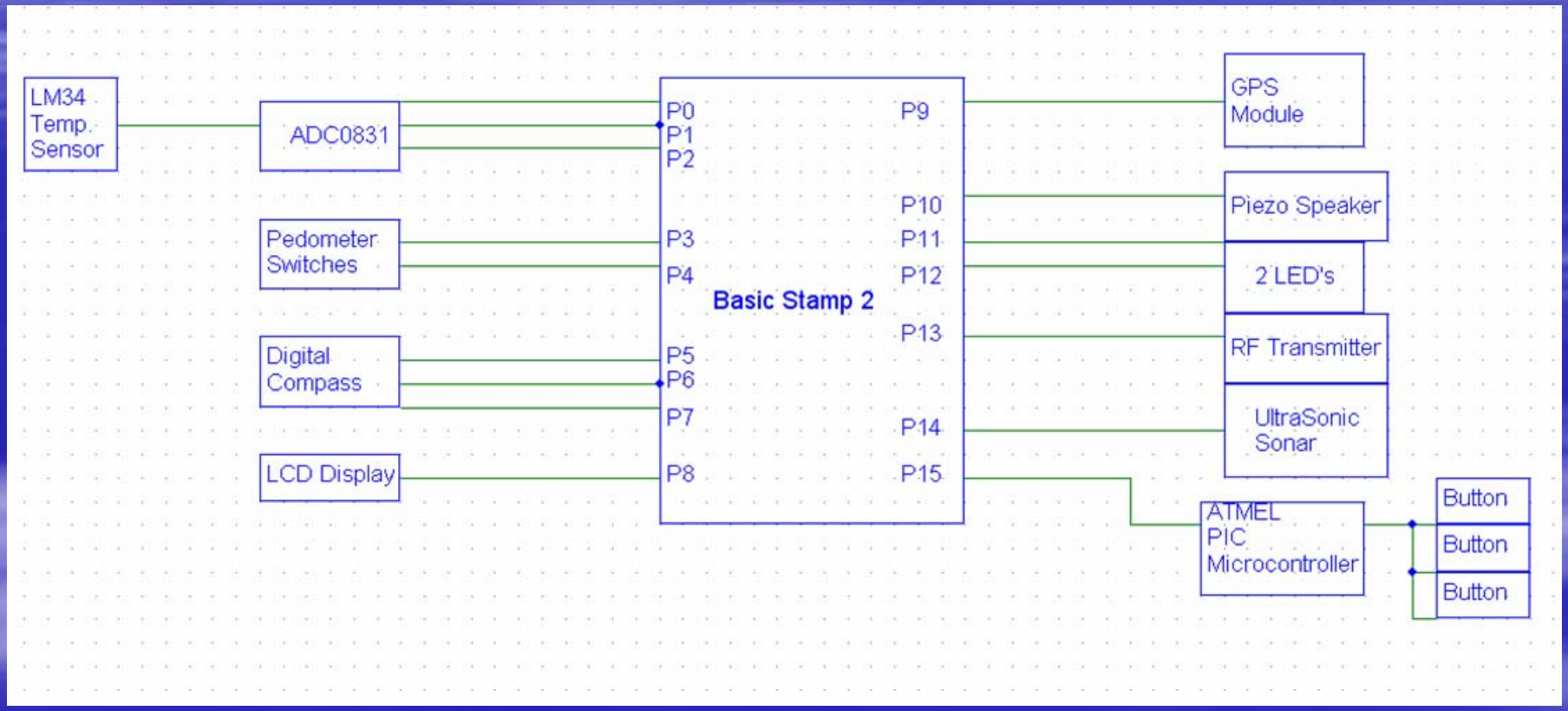
- C-chip ATML Pic Microcontroller

  Used to offload tasks from the Basic Stamp

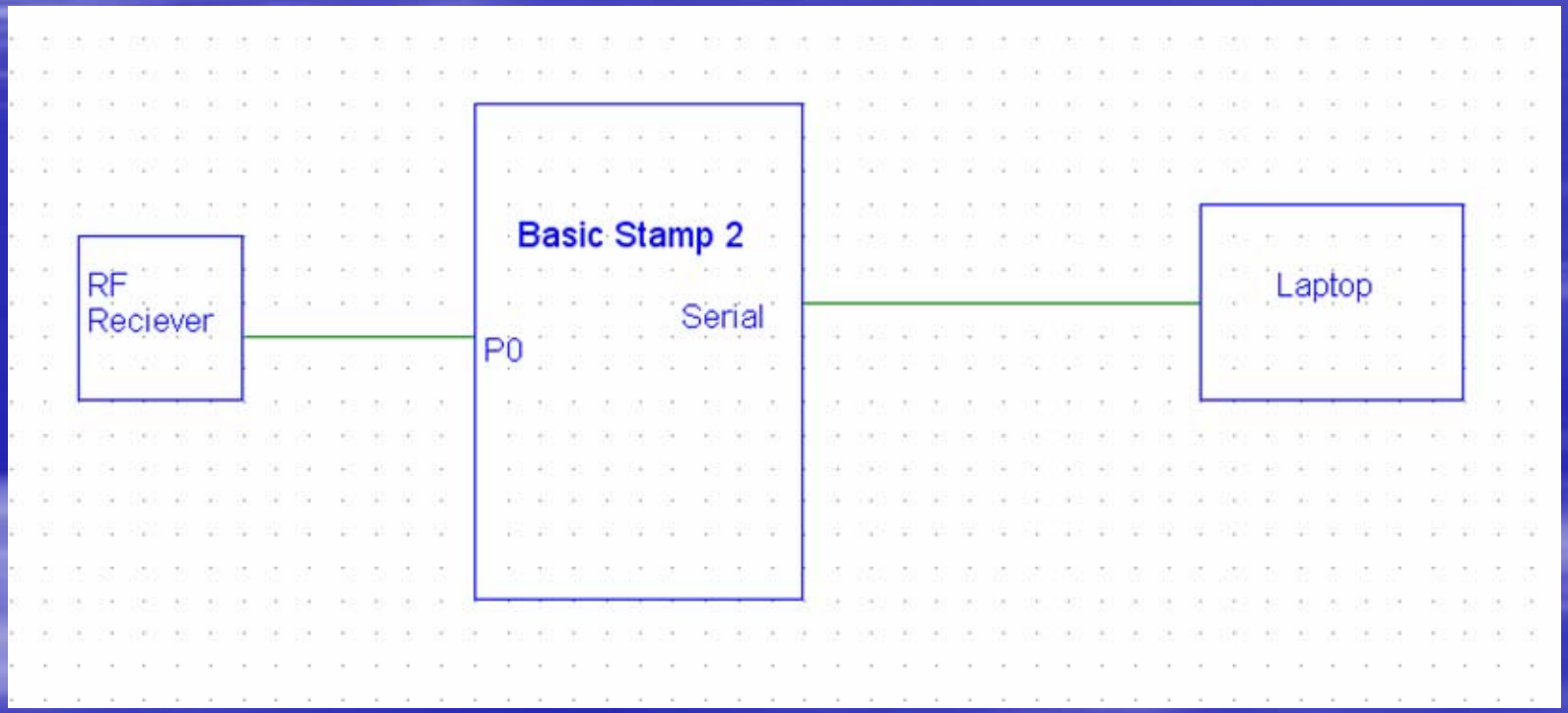
Hardware V

- LCD + Buttons for User Interface
  - 2 x 16 LCD display to show user position and vital information
  - 3 push buttons for different sets of data
- Emergency Shut Down Switch
  - In place for emergencies and unsafe operation conditions.

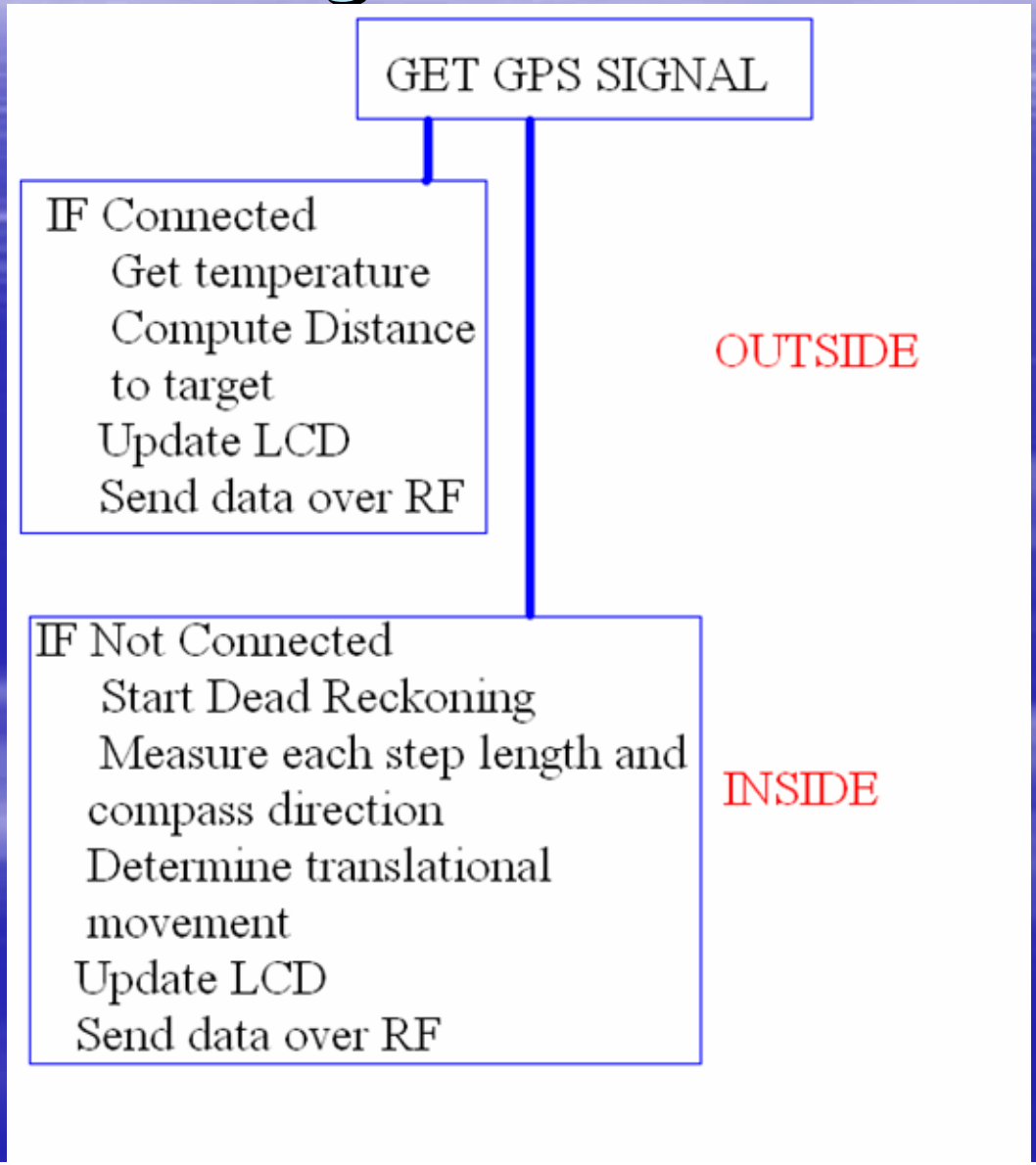
Schematic


LM34
Temp.
Sensor
ADC0831
Pedometer
Switches
Digital
Compass
LCD Display
P0
P1
P2
P3
P4
P5
P6
P7
P8
Basic Stamp 2
P9
P10
P11
P12
P13
P14
P15
GPS
Module
Piezo Speaker
2 LED's
RF Transmitter
UltraSonic
Sonar
ATMEL
PIC
Microcontroller
Button
Button
Button

Schematic



Coding Flow Chart

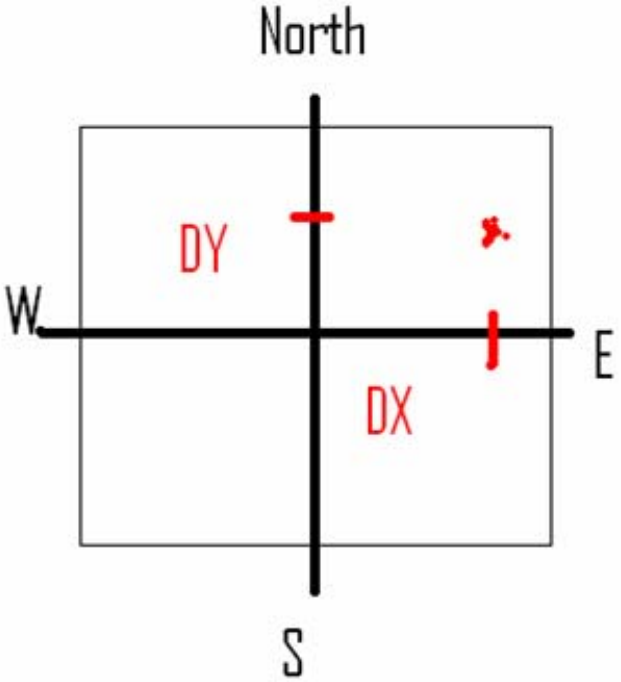


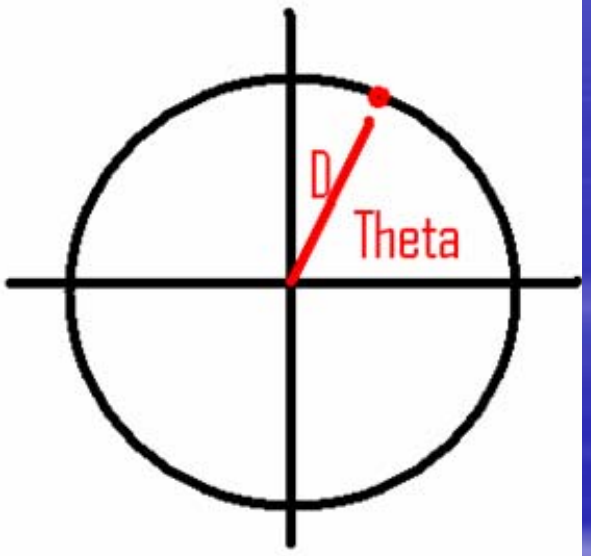
Code

```
IF(buttonchoice = 3) THEN GOSUB LCDHELP
'encryption
lat1 = lat1 + code
lat2 = lat2 + code
long1 = long1 + code
long2 = long2 + code
SEROUT rf_transmitter_pin, 16468, ["$GPGGA,", DEC code, DEC lat1, DEC lat2, DEC long1, DEC long2,
GOSUB GETCOMPASSANGLE
GOSUB UPDATESPEAKER
GOSUB UPDATEGOGGLES
RETURN
'''''''''''''''''''''''''''''''''''''''''''''
GETGPSLATLONG:
SEROUT gps_pin, 188, ["!GPS", GetLat]
SERIN  gps_pin, 188, 3000, DEADRECKONING, [lat1.HIGHBYTE, lat1.LOWBYTE, lat2.HIGHBYTE, lat2.LOWBYT
WRITE 0, Word lat1
WRITE 2, Word lat2
SEROUT gps_pin, 188, ["!GPS", GetLong]
SERIN  gps_pin, 188, 3000, DEADRECKONING, [long1.HIGHBYTE, long1.LOWBYTE, long2.HIGHBYTE, long2.LO
WRITE 4, Word long1
WRITE 6, Word long2
  'DEBUG lat1.HIGHBYTE, " ", lat1.LOWBYTE, ".", lat2, " - ", long1.HIGHBYTE, " ", long1.LOWBYTE, "
RETURN
''''''''''''''''''''''''''''''''''''''''''''''''''
DEADRECKONING:
'DEBUG CRSRXY, 6,6, "DEADRECKONING", CR
IF(firsttime=0) THEN
  firsttime=1
  lat1=0
  lat2=0
  long1=0
  long2=0
  temporary=0
  temperature=0
  pedometer_steps=0
  adc=0
ENDIF

IF(IN4=0) THEN down1=0
IF(IN3=0) THEN down2=0
IF(IN4=1) AND (down1=0) THEN GOSUB right
IF(IN3=1) AND (down2=0) THEN GOSUB left
IF(pedometer_steps >=15) THEN GOTO MAIN
'DEBUG CRSRXY, 3, 3, "getting button choice", CR
GOSUB GETBUTTONCHOICE
IF(buttonchoice = 1) THEN GOSUB LCDDXDY
GOSUB GETCOMPASSANGLE
GOSUB WAYPOINT
GOSUB GETTEMPERATURE
IF(buttonchoice = 2) THEN GOSUB LCDTEMPERATUREDR
IF(buttonchoice = 3) THEN GOSUB LCDHELP
'GOSUB RFDEADRECKON
```

Math





Theta = arctan (y / x) OR ATN

D = sqrt(dx^2 + dy^2) OR HYP

Know: Latitude, Longitude, compass angle and distance travelled

Cost of Materials

Part	Amount	Cost/Unit	Price	Mass. Unit Price
Basic Stamp	2	49.99	$99.98	$25.48
GPS Sensor	1	79.95	$79.95	$71.96
RF Reciever	1	39.95	$39.95	$27.97
RF Transmitter	1	29.95	$29.95	$20.07
Compass	1	29.95	$29.95	$23.96
LCD	1	29.95	$29.95	$26.96
Sonar	1	29.95	$29.95	$29.95
Temperature Probe	1	7.90	$7.90	$7.90
Miscellaneous			$48.04	$10.13

Total	$395.62

Future Plans/Improvements

- Basic Stamp with more variable space and I/O pins
- More Accurate Pedometer

  -Redundant Motion Recognition
- Better Mounting
- Weather proofing
- Longer range RF transmitter

Demo

References

- References

Parallax – www.parallax.com

Radio Shack – www.radioshack.com

Google Image Search: Step Length – www.google.com

Google Maps – http://maps.google.com

Pspice Electronics Software – www.pspice.com

Python programming language – www.python.org

Wikipedia Future Combat Systems
http://en.wikipedia.org/wiki/Future_Combat_Systems