Project Title

Automated Guided Vehicle
Project Goals

• To develop and construct a prototype of a low cost, user friendly, automated guided vehicle (AGV) to contribute to an efficient material handling system in a small-scale industry.

• To communicate wirelessly to and from the AGV from the base station.

• The AGV would be guided with the help of IR emitters planted in the ground that would be detected by the on-board IR receivers.

• Instructions sent to the robot has to be executed and synchronized with the base station.
Construction

- Standard Components

- BOE – Bot Kit [From Parallax]
  - Servo
  - LCD
  - IR Receiver

- IR Emitters, Resistors, Diodes and Wires [Radio Shack]
Working - I

• The AGV is retrieved from the charging dock and sent to the first workstation.

• IR emitters fitted in the ground define the path to the first workstation.

• At the workstation the AGV awaits instructions from the user.

• IR emitters define the next path to be taken by the AGV to the next workstation.
Working - II

- The path can be defined by entering the workstation number and BS2 will reflect the shortest route by exciting the corresponding IR emitters.

- The shortest route is predefined by diode logics.

- The AGV will move to the next workstation and await instructions.

- At the end of the shop floor shift, the AGV will be triggered to return to its charging dock.

- As a safety precaution, the AGV would stall in place if there were an obstacle ahead of it.
Key Features of the Project

• Aimed at developing a low cost material handling equipment.

• Incorporates wireless devices to send/receive information wirelessly from the BS2 on the AGV.

• The path is defined using logics developed using electrical components that are controlled by another BS2.

• Incorporating safety feature to stall immediately.

• Components of factory standards have been incorporated.
Cost Analysis

• Cost to make 1 AGV - $373.12
• Cost when mass produced - $285.09
• Savings – 23.59%
Robot Logic

CASE 2, 8
pulseleft=768
pulseright=731
robot_state=1
DEBUG "Moving forward", CR
SEROUT 3, 84, [DEC1 next_station, 13, "Moving forward"]
CASE 14

IF robot_state>0 THEN
  pulseleft=768
  pulseright=731
  'SEROUT 3, 84, [22, 12]
  'SEROUT 3, 84, [DEC1 next_station, 13, "Moving forward"]
ENDIF
CASE 12
'turn left
pulseleft=731
pulseright=731
robot_state=2

SEROUT 3, 84, [DEC1 next_station, 13, "Turning left"]
DEBUG "Turning left", CR
Robot Logic

CASE 6

'turn right
pulseleft=768
pulseright=768
robot_state=3
DEBUG "Turning right", CR
'SEROUT 3, 84, [22, 12]
SEROUT 3, 84, [DEC1 next_station, 13, "Turning right"]
CASE 10
  pulseleft=750
  pulseright=750
  IF robot_state>0 THEN
    'SEROUT 3, 84, [22, 12]
    SEROUT 3, 84, [DEC1 next_station, 13,
    "Stopping..."]
  ELSE
    SEROUT 3, 84, [22, 12]
    SEROUT 3, 84, [DEC1 next_station, 13,
    "Stopped"]
  ENDIF
  robot_state=0
  DEBUG "Stopping", CR
  curr_station=next_station
  GOTO Arrived
ENDSELECT
Board Logic

NEtoNW:

DO
  FREQOUT 5, 5, 38500
  SERIN BaseIN, 84, 20, NEtoNW, [curr_station]
  LOOP UNTIL curr_station=next_station
  GOTO Confirm
NEtoSE:
DO
'Light up the correct path
FREQOUT 4, 5, 38500
SERIN BaseIN, 84, 20, NEtoSE, [curr_station]

LOOP UNTIL curr_station=next_station '
DEBUG "Arrived at: ", DEC curr_station, CR
GOTO Confirm
Problems Incurred

- Leak in RF Emitters
- Detection problems in RF Receivers
- Timing difference between the emitter and the receiver
- RF Transceivers circuitry
- SERIN / SEROUT
- Breadboard Problems
- Grounding Problems
DO
   FREQOUT 5, 5, 38500
   SERIN BaseIN, 84, 20, NEtoNW, [curr_station]

LOOP UNTIL curr_station=next_station
   GOTO Confirm
Scope

- Can be integrated to SAP.
- Modules such as Wi-Fi and GPS can be added.
- Attachments such as end effectors can be added.
- High powered Microprocessor can be used.
- Can be extended to the entire industry having many units.
QUESTIONS?
Demo!