



AGV Photo courtesy of Artisteril SA. Barcelona. Spain

An automated guided vehicle (AGV) can be built with as little as two Roboteq components: An MGS1600C magnetic guide sensor, and one of the many dual channel motor controllers available in Roboteq's catalog. This application note will use a Roboteq MDC2260 dual channel controller, however the techniques described are identical for all other controller models.



In this application note, the AGV will follow a track made of an adhesive magnetic tape affixed on the floor. The MGS1600C will measure the how far from the center of the tape it is and provide the

information to the motor controller which will then adjust the steering so that the vehicle remains at the center of the track.

Magnetic markers positioned on the left and right side of the track give the AGV location information that will be used to make stop and fork left/right decisions.

Benefits of Magnetic Track Guiding

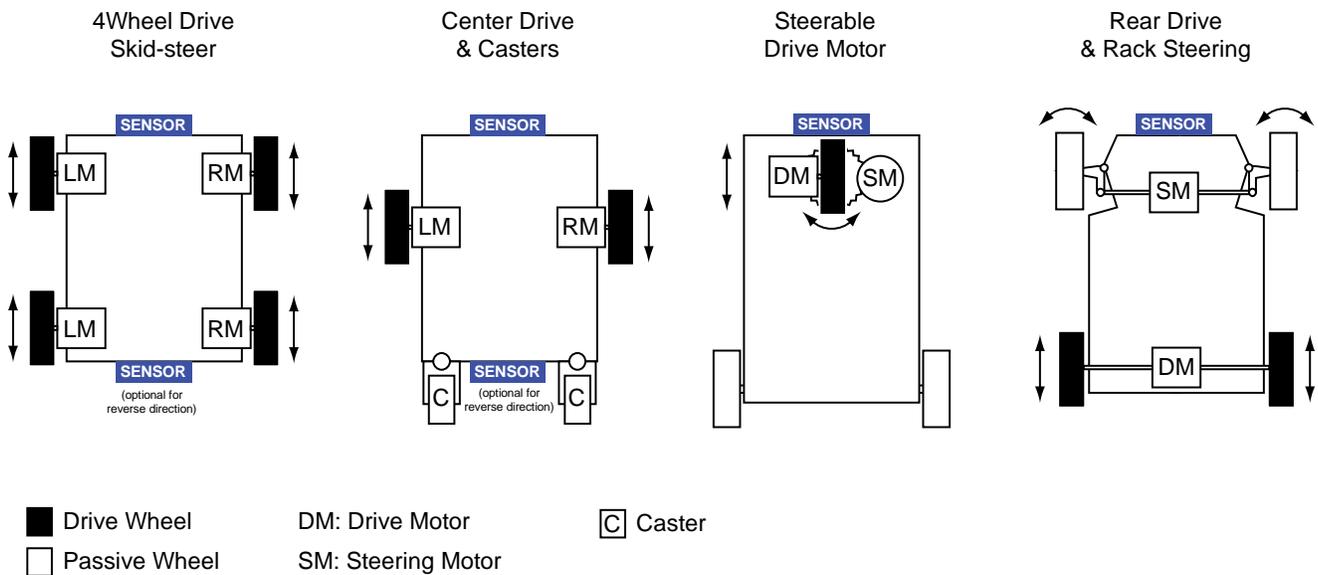
Magnetic tapes are one of several line-following techniques. The other 2 main techniques are induction wire guide, and optical. The table below compare each of these techniques

Technique	Easy to lay	Easy to change	Passive	Dirt immune	invisible
Magnetic Guide	Yes	Yes	Yes	Yes	Yes
Induction wire	No	No	No	Yes	Yes
Optical	Yes	Yes	Yes	No	No

Magnetic tape is the easiest to lay and modify while providing excellent durability and reliability.

Chassis Design

When designing the vehicle, there are four basic ways of providing drive and steering. These are show in the diagrams below. Some types are easier to build, others have better steering characteristics. Two of these designs are fully symmetrical and may be operated in both direction. The table gives list the characteristics of each design.



Design	Simplicity	Steering	Reversible
4Wheel Drive	Simplest	Coarse	Yes
Center Drive & Casters	Simple	Precise	Yes
Steerable Drive Motor	Medium	Very Precise	No
Rear Drive and Rack Steering	Complex	Precise	Difficult in reverse

Sensor Mount

The sensor should be placed as shown in the above diagrams for each chassis design. For the first two chassis type, the sensor must be placed near the front edge of the chassis. On long AGVs, this means that little steering will cause a wide swing at the front and will make the steering control more difficult.

On the steerable drive wheel design, the sensor can be placed on the chassis. Or it may be made part of the wheel assembly and turn with it.

For best results, place the sensor at 30mm above the floor and ensure that the height fluctuates within +/- 10mm max as the AGV moves along the track.

Interfacing the Sensor to the Motor Controller

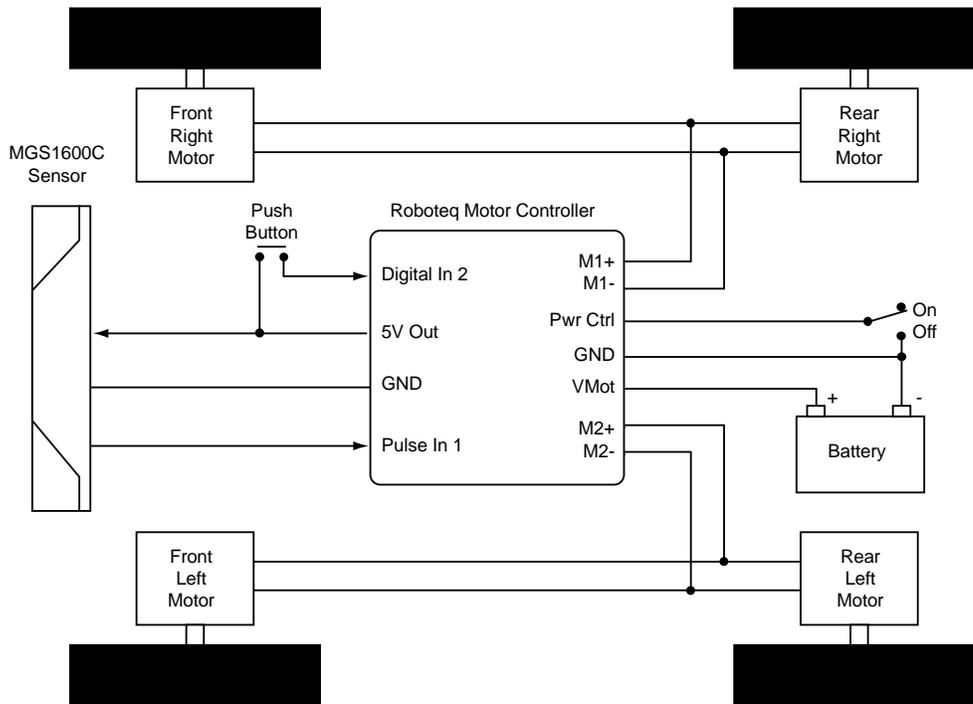
The MGS1600C has several output types. The list, features and typical uses is shown in the table below.

Output	Sensor Data Sent	Noise immune	Use
RS232	All	Yes	PLC, Computers
Multi PWM	Position, Tape Detect, Markers	Yes	Roboteq motor controllers
Analog	Position only	No	PLC
PWM	Position only	Yes	PLC
CAN	All	Yes	PLC, Computers
USB	All	Yes	PC Motherboards

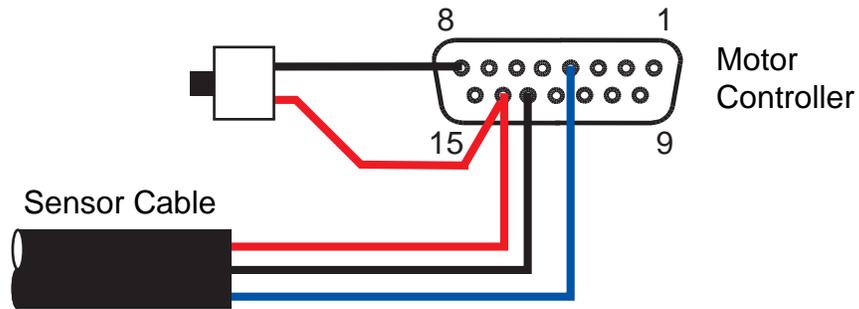
In the MutliPWM mode, the sensor data is output on a single wire in the form of a series of variable widths pulses, containing the Track Detect signal, Track Position and Left & Right Marker Detects signals. This pulse can be connected to any of the Roboteq motor controller's pulse inputs. Once the pulse input is configured as "Magsensor", the sensor information is transferred transparently and continuously to the motor controller, from where it can be processed using the MicroBasic scripting language, or accessed by an external computer or PLC via the controller's serial or USB port

Electrical Wiring

The wiring diagram below shows the magnetic guide sensor and motor controller in a typical 4 wheel drive chassis. This diagram is applicable to all Roboteq dual channel brushed motor controllers.



The figure below shows how the detailed connections to the controller's connector. This wiring is compatible with all Roboteq controllers equipped with a 15-pin DSub connector. The sensor and button can be connected to any other pulse and digital inputs. Refer to the product datasheet for the list of available signals and pin out. The pulse output in on the blue wire of the sensor cable.

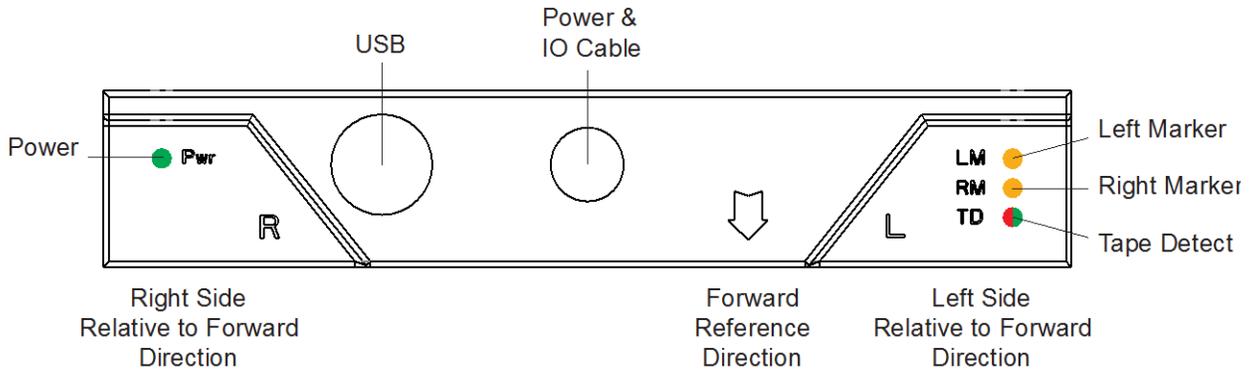


Configuration and Testing

The sensor and controller must be configured so that they will each function as desired, and that they will communicate with each other.

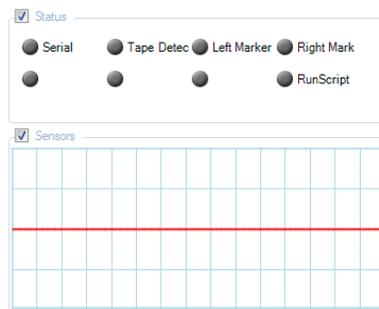
Sensor Configuration and Testing

The sensor is configured by default to output MultiPWM pulse and can therefore be used without further configuration if the track is made of Roboteq-supplied 25mm magnetic tape. Once powered, the Tape Detect LED will flash at a low rate if no tape is present. When tape is in range, the LED will be on steady and its color will change from green when the tape is at the right, and green when at the left

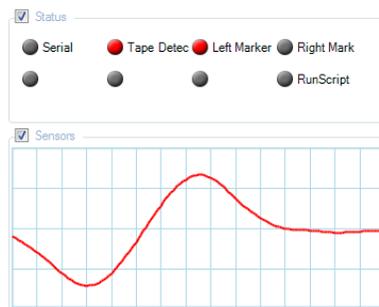


For configuration, monitoring and troubleshooting, connect the sensor to the PC via the USB connector located under the screw plug. Run the Magsensor PC utility to change the tape width if using a 50mm tape, or use the waveform display view to monitor the shape of the magnetic field.

With no tape present, the PC utility must show a nearly flat line. For best results, always perform a zero calibration when operating the sensor in a new environment.



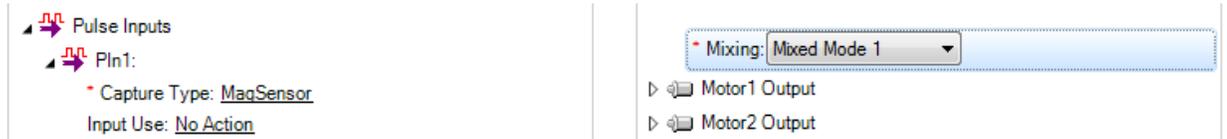
Moving a tape under the sensor will cause a "bell" curve to appear on the chart. The curve must be on the positive (up) direction. If the curve is going down, change the Tape Polarity setting in the configuration menu. Makers will also cause a bell curve, but the curve should be going down. The screenshot below shows the resulting curve when placing a left marker and a centered track.



Motor Controller Configuration

To receive and recognize data from the sensor the controller must first be connected to a PC running the Roborun+ PC utility. In the configuration menu, the pulse input that is connected to the sensor must be enabled and configured as "Magsensor".

Next, the controller must be configured to operate in mixed mode so that a steering command will apply a different amount of power to the left and right motor for making turns.



In the Run tab of the PC utility, the sensor can be seen to work when the Digital Input 1 LED is randomly flickering. The Pulse 1 box will display the number 128 if no tape is detected and a different value as a tape is moved from side to side.

Magsensor to Controller Interface

When the sensor and the motor controller are connecting to each other using the single wire and MultiPWM mode, the sensor data are transferred periodically, and in the background, into the motor controller from which they can then be accessed and used.

An additional set of queries is now available for reading this information from the motor controller. The queries can be sent either from the motor controller's serial port, or from within a MicroBasic script running in the motor controller.

Serial/USB Query	MicroBasic Alias	Description
?MGD	_MGD	Read Tape Detect
?MGT	_MGT	Read Left or Right Track position
?MGM	_MGM	Read Left of Right Marker
?MGS	_MGS	Read Sensor Status. Non-zero = sensor present and working

Implementing the AGV Controls

Once the sensor and motor controllers are verified to work, we can proceed to the automatic mode. In this application note, all the computation is done in the motor controller using the MicroBasic scripting language

Steering Control

The sensor outputs a value that is the tape's distance from the center of the track. This information is then used to correct the steering. If the tape is centered, the value is 0 and no steering correction is needed. The further the track is from the center, in one or the other direction, the stronger the steering change. In this article, a Proportional control is implemented. For best precision and response time, the control algorithm may be improved to a full PID.

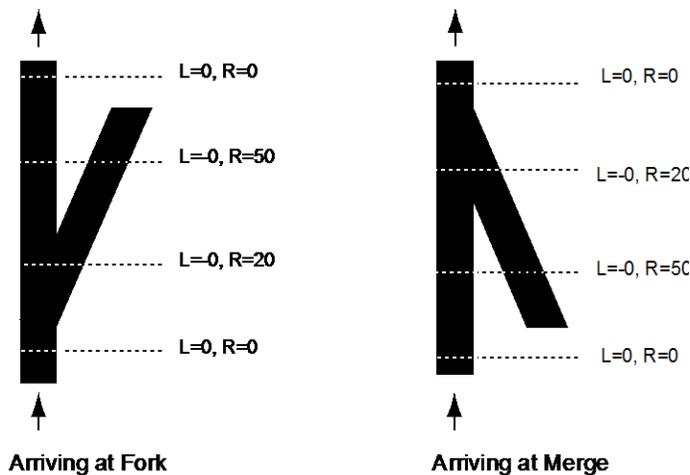
Throttle Control

How the throttle power is controlled (when to start, stop, accelerate, slow down) is very application dependent. In this application note, the AGV will be made to move when a tape is detected, take left or right forks and stop at precise locations. The AGV will then resume moving after a set time, or when a user button is pressed. The AGV will stop when the track is no longer present.

In a practical implementation, the AGV throttle will be controlled by an external device, such as a PLC. The PLC must then be connected to one of the motor controller's input. The throttle information can be an analog voltage or a variable duty cycle PWM signal.

Fork and Merge Management

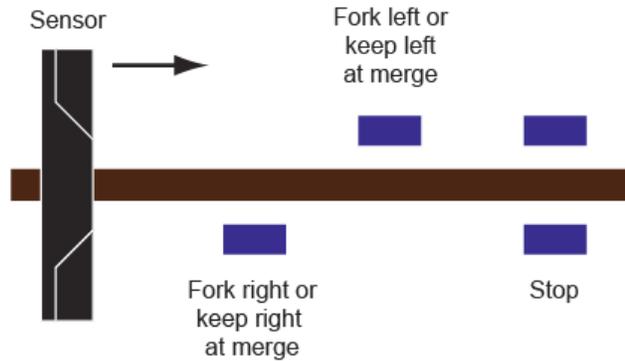
The sensor has an algorithm for detecting and managing up to 2-way forks and merges along the track. Internally, the controller always assumes that 2 tracks are present: a left track and a right track. When following a single track, the sensor considers that the 2 tracks are superimposed. When entering forks, the track widens, so does the distance between the left and right tracks. When approaching merges, the sensor will report a sudden spread of the left and right tracks, but will otherwise operate the same way as at forks



Localization using Markers

Magnetic markers are a piece of magnetic tape of opposite polarity and that is located left and/or right of the center track. Markers provide a very simple and cost effective method to identify specific locations along the track.

In this application, we will use markers on the left or right side to indicate which track to follow at a fork. Markers located both at the left and right side will indicate a stop location.



More elaborate marker arrangement can be made to carry more information about a location on the track. An example of multi-level markers is provided further in the application note.

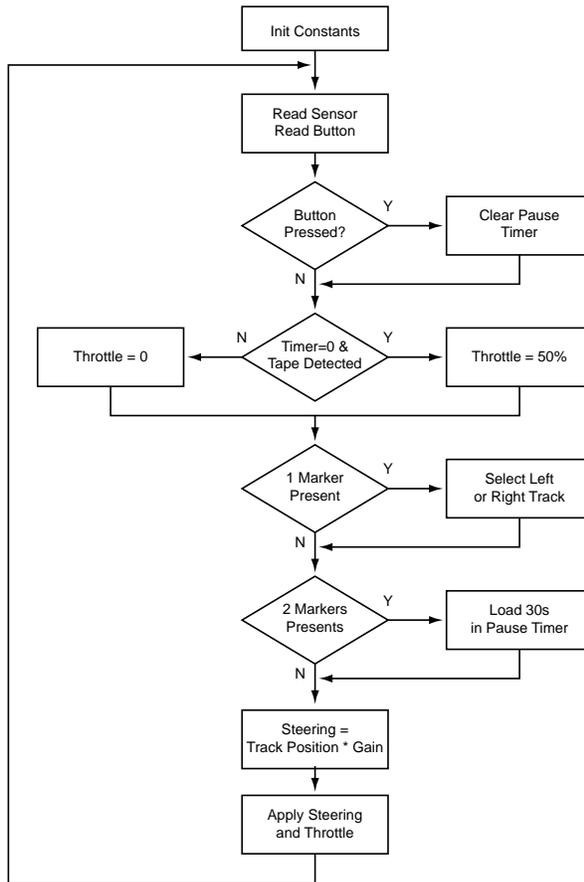
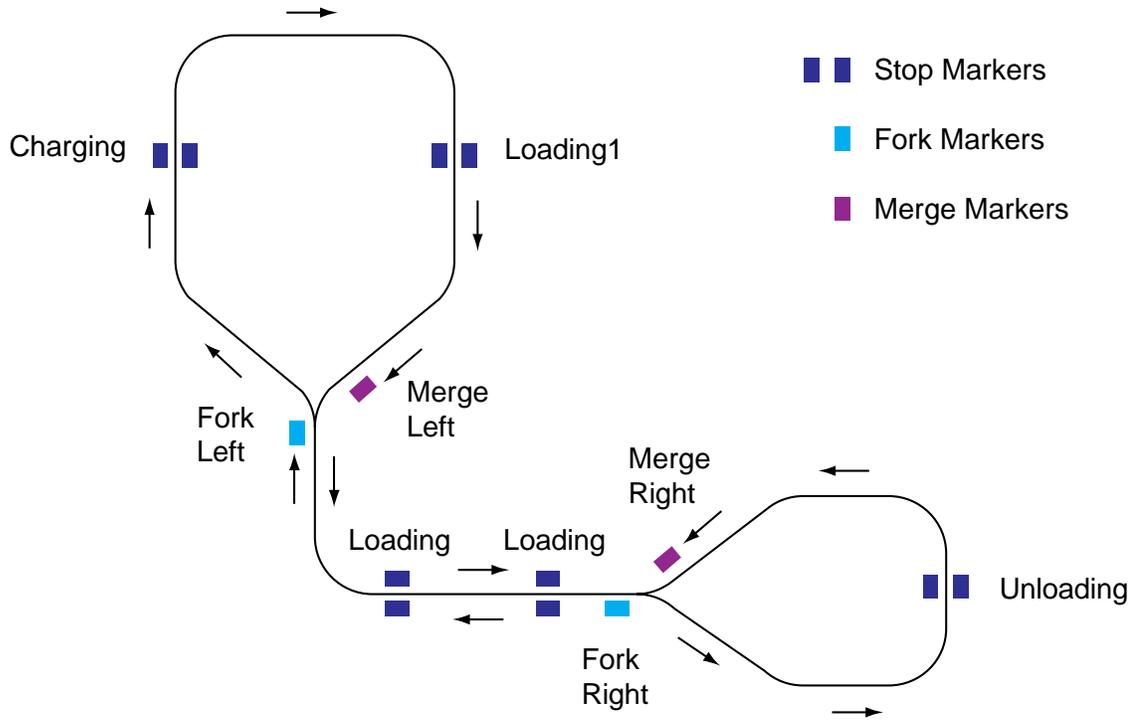
Manual Steering Override

It is common to require that the AGV be driven manually, to place it in position, or to move it along an untracked path. Buttons, a joystick, a PLC or an RC Radio can be connected directly to the motor controller's free inputs. The program running inside the motor controller can easily be made to switch from automatic to manual command. Manual override is not described in this Application Note.

Test Track Description

The figure above shows a simple AGV track with several loading station and one stop station. For simplicity sake, our AGV will stop 30 seconds at every station, or until the operator presses the push button.

The flow chart below shows the structure of the MicroBasic program that will run inside the motor controller to move and steer the AGV along the track. The full source code is provided at the end of this application note.



Setup and Troubleshooting

Manually Test Drive the AGV

Before the sensor can be used for automatic steering, it is a good idea to test drive the chassis manually, either by attaching a joystick to the PC that is connected to the motor controller, or by using an RC radio. If the vehicle is difficult to drive manually, in automatic mode it will be equally challenging. Modify the design so that it drives and steers as smoothly and accurately as possible

Testing the Automatic Steering Program

When running the program for the first time, it is recommended to lift the AGV's wheels off the ground. Then place a piece of magnetic tape below the sensor. Verify that the left and right wheels start rotating when the tape is detected. Verify that the left and right rotation speed changes as the tape is moved away from the sensor's center, in a manner that would cause the AGV to rotate so that the sensor would move become centered with the tape. If the AGV rotates away, then invert the polarity of the Gain value in the script.

With the AGV wheels on the floor, verify that the steering correction is such that the sensor never moves far from the track. Increasing the Gain value will cause a stronger correction when the sensor moves away from the tape, but it can make the AGV oscillate if the gain is too high. Find the optimal Gain value for stable and accurate tracking.

Testing Fork and Stop Markers

Markers are best tested with the AGV on the track. Verify that the AGV follows the expected track at a fork. When entering a merge, ensure that the AGV is following the correct track and that it will not jump to the opposite track when it enters the sensor's range.

Verify that the AGV stops when a left and right marker are detected at the same time. Check that the AGV resumes motion after 30 seconds, or when the button is pressed. Beware that as the AGV moves away from the marker pair, one of the two markers will disappear from the sensor's range before the other. The other marker is remain active for short duration longer and will therefore be considered as a left or right fork marker. Make sure, therefore, that a single marker is present before the next fork or next merge following a stop location.

Improving the AGV

Using a more complex steering algorithm

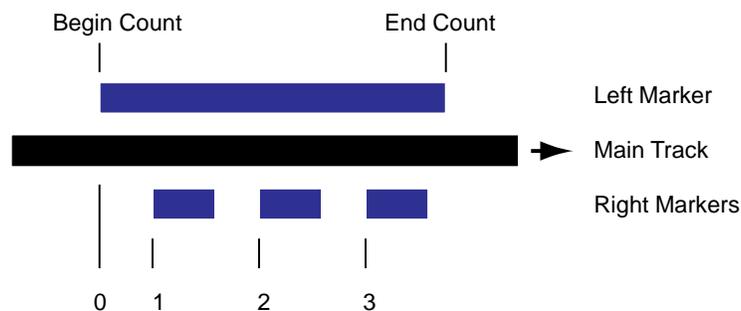
The sample script uses a simple Proportional control, where the amount of steering correction is simply the distance away from the center track, multiplied by a gain factor. For better results, the script may need to be enhanced so that a Proportional-Integral or full Proportional-Integral-Derivative control is used instead.

The amount of correction can also be capped to avoid over steering. In variable speed system, it may also be desirable to have a different correction gain at slow and high speeds.

Using Multi-Level Markers

In this application example, we used a simple left and right marker pair to identify a stop location. In typical applications more information is needed about location so that the AGV will change its behavior. For example, identifying segments of tracks where the AGV must move at a high speed and others at low speed, identifying load stations requiring longer pause time than others, or identifying charging stations where the AGV will only stop when its battery level is low and resume when the battery is charged.

One simple and free technique is to count markers in a track segment delimited by a marker segment on the opposite side. The figure below shows such a marker configuration. When a left marker first appears, the counter is reset. The counter is then incremented at every appearance of a right marker while the left marker is still present. When the left marker disappears, the counter is evaluated and the AGV can alter its operation accordingly.



The MicroBasic source code for creating this marker is shown below.

```

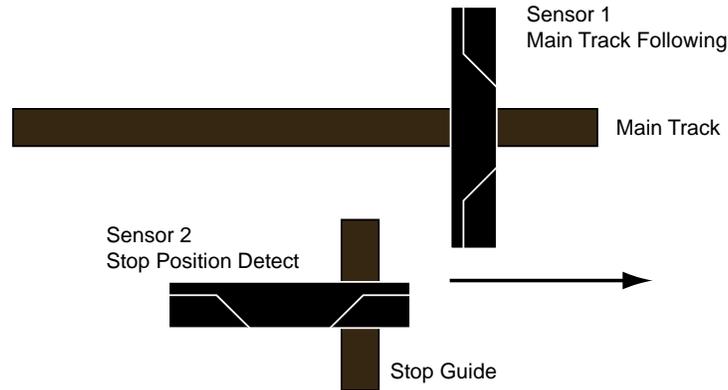
if(Left_Marker = true) ' When left marker is present
  if (PrevLeftMarker = false)
    MarkerCount = 0      ' clear counter when left marker first appears
  elseif(Right_Marker = true and PrevRightMarker = false)
    MarkerCount++      ' inc counter at every appearance of right marker
  end if

elseif (PrevLeftMarker = true) ' Left marker is absent but was present previously
  if (MarkerCount = 1)
    ' enter code for one count marker here
  elseif (MarkerCount = 2)
    ' enter code for two count marker here
  elseif (PrevLeftMarker = 1 and EventCount = 3)
    ' enter code for three count marker here
  end if
end if

' Save markers for change detection at next cycle
PrevLeftMarker = Left_Marker
PrevRightMarker = Right_Marker
    
```

Improving Stop Position Accuracy

In applications requiring the AGV to stop at a very precise location, a secondary sensor, oriented at 90o from the main sensor, can be added. This sensor can then be used to locate another magnetic guide with a 1mm position accuracy. The figure below shows the sensors and guides arrangements.



Improving AGV Localization and Safety

For safety reason, it is typically necessary to fit the AGV with an infrared or a laser range finder so that it will stop if a person or obstacle is detected along the track. Range finders typically provide a digital signal which can easily be connected to an input on the motor controller, or to a PLC if one is present.

If more information is needed by the AGV about its location along the track, RFID tags positioned in key locations are a good solution. However, RFID tags typically imply the presence of a microcomputer or a PLC on the AGV in order to process the data and make navigation decisions.

Script Source

The source code below is written in Roboteq's MicroBasic language and runs inside the motor controller to perform the AGV functionality described in this application note.

```
option explicit

' This script provide basic control for an AGV.
' Motor will turn on upon the presence of a track and stop when track disappears.
' The track position information is used to provide left/right steering.
' At forks, the AGV will follow the left or right track depending whether the last
' marker detected was on the left or right side of the track.
' Make sure you precede merges with a marker so that the AGV remains on the main track.
' The presence of a left and right marker simultaneously will cause the AGV to stop for
' 30 seconds or until the operator presses the button

' declare variables
dim Gain as integer
dim DefaultThrottle as integer
dim TapeDetect as boolean
```

```
dim MarkerLeft as boolean
dim MarkerRight as boolean
dim Throttle as integer
dim Tape_Position as integer
dim LineSelect as integer
dim Steering as integer
dim GoButton as boolean
dim RunState as boolean
dim NotOnStopMarker as boolean
dim PauseTime as integer

' initialize constants
Gain = -7 ' Use negative value to invert steering command
DefaultThrottle = 250 ' Motor power level while the AGV runs
LineSelect = 1 ' Use left track by default
PauseTime = 30000 ' in miliseconds

' main loop to repeat every 10ms
top:
wait(10)
' read sensor data
TapeDetect = getvalue(_MGD)
MarkerLeft = getvalue(_MGM, 1)
MarkerRight = getvalue(_MGM, 2)

' Read button state
GoButton = getvalue(_DI, 2)

if (GoButton) then SetTimerCount(1, 0) ' Pressing the button will clear the pause timer
if GetTimerState(1) then RunState = true ' When pause timer is cleared, AGV is allowed to run

' Use TapeDetect and Pause Timer to apply throttle or not
if (TapeDetect and GetTimerState(1))
    Throttle = DefaultThrottle
else
    Throttle = 0
end if

' Check Marker presence to select Left or Right track
if (MarkerLeft) then LineSelect = 1
if (MarkerRight) then LineSelect = 2

' Detect when transitioning onto stop markers
if (NotOnStopMarker and MarkerLeft and MarkerRight)
    NotOnStopMarker = false ' Mark stop marker detection so that it is not detected again
until AGV moved away
    SetTimerCount(1, PauseTime) ' Load stop timer timeout value
    RunState = false
else
    NotOnStopMarker = true
end if

Tape_Position = getvalue(_MGT, LineSelect)

' use tape position multiplied with gain as steering
Steering = Tape_Position * Gain

' Send throttle and steering to controller Configured in Mixed mode
setcommand(_G, 1, Throttle)
setcommand(_G, 2, Steering)

' Log output. Useful for troubleshooting. Comment out when done.
print("\r", TapeDetect, "\t", Tape_Position, "\t", MarkerLeft, "\t", MarkerRight, "\t",
Throttle, "\t", Steering, "\t", RunState, "\t", LineSelect)

goto top ' Loop forever
```