Remote-IO Provides A Flexible Interface to Sensing and Actuation

Abstract

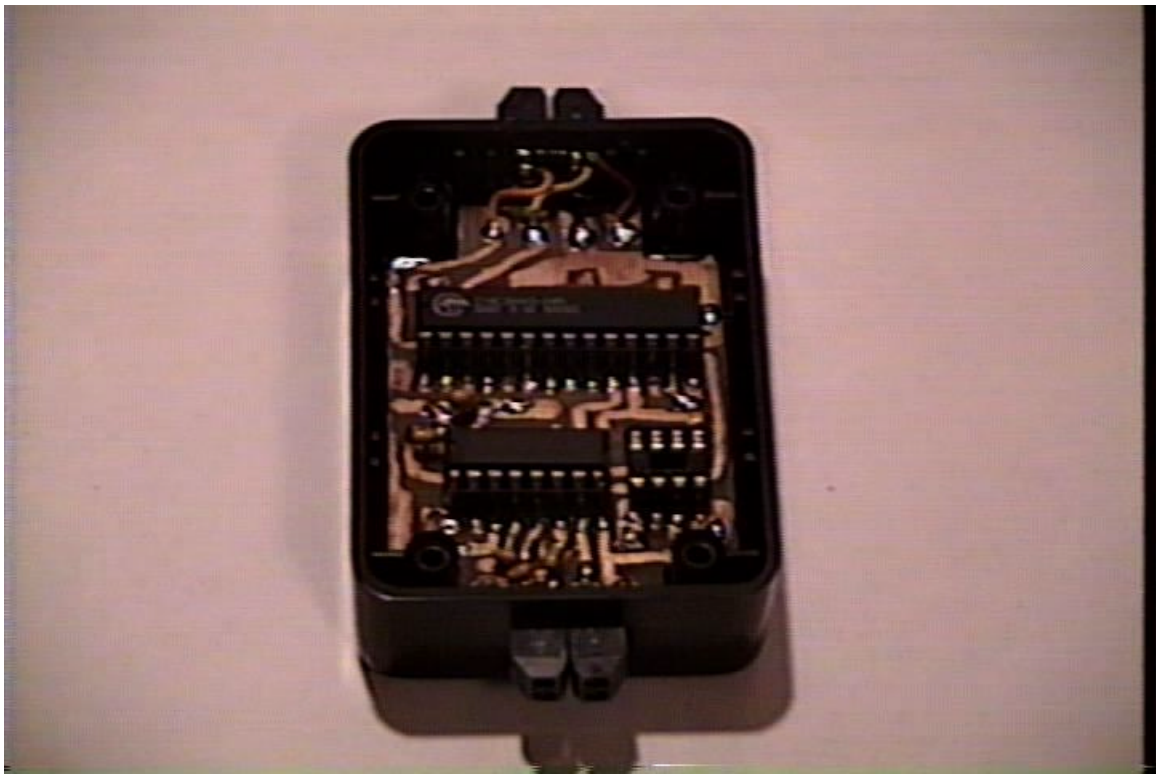
Sensing and actuation are key elements in embedded machine control. In an industrial setting, large manufacturing machines require sensors and actuators distributed over large physical spaces. Discrete wiring to each of these devices can be cumbersome and error prone. An alternative solution is an interface for sensors and actuators that can be daisy-chained around the machine. New devices can be added or removed at any location without changing the wiring back to the controller. Remote-IO provides a way to implement this system.

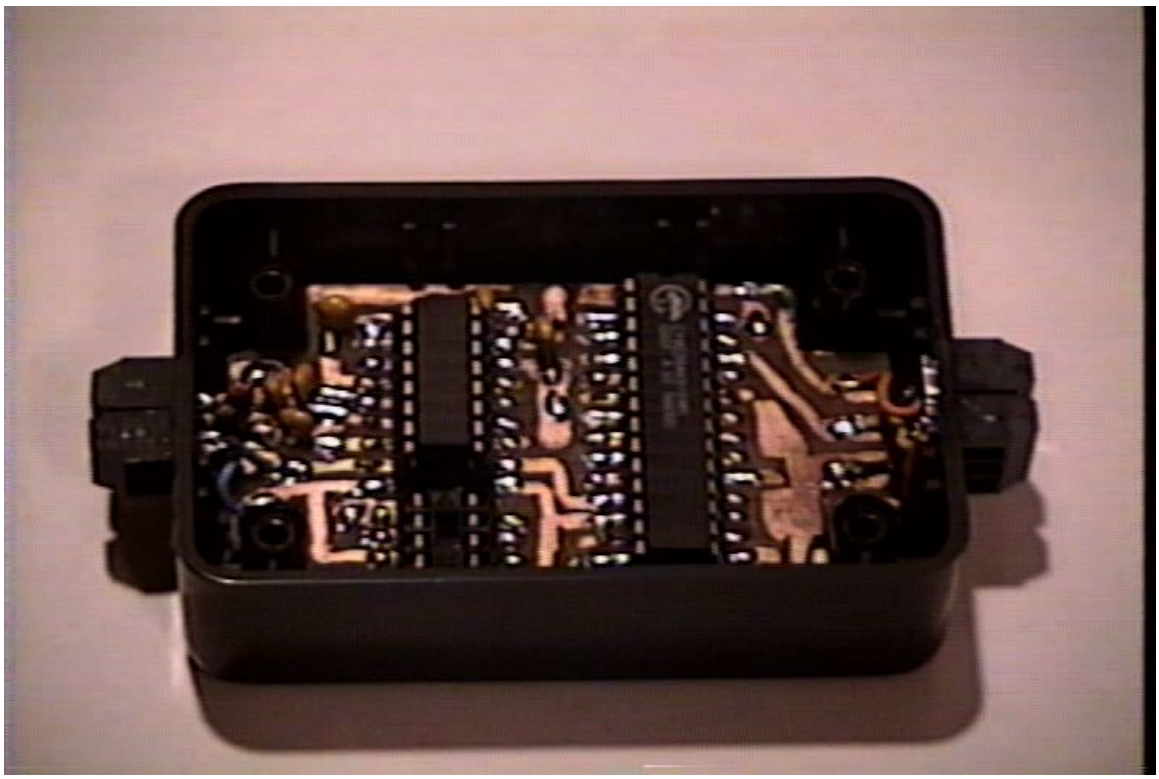
Each Remote-IO device consists of a multi-purpose, multi-point input/output node. Each node has two independent interface pins to the environment to be sensed and controlled, and a serial communications interface back to the host controller. Each control pin can be configured as either a digital input, a digital output, a ADC input, a DAC output, an digital input event counter, a timed-on digital output, or a PWM output.

This device can be implemented with a minimum of parts by using the Cypress MicroSystems PSoC 8C26443 microcontroller. Not only is the part count kept low, but the layout of the parts is simplified because the assignment of internal function to the microcontroller's pins is flexible. In the block diagram it can be seen that the device consists of the microcontroller and a serial interface chip. In this implementation of the prototype, the choice of RS232 or RS485 is made by inserting the appropriate driver into the socket. RS232 is a natural choice for a prototype because of the universal availability on PC's. RS485 is the best choice for a large number of devices on a distributed machine network. The block diagram also shows the rich selection of functions available to each of the two sensor/actuator pins. Selection of pin function is made by a serial command sent to the device. The device echoes the command and, if required, appends any data requested to the return message. Development of the prototype was aided by the flexibility of the design. In order to confirm the operation of a particular function, a complementary function can be assigned to the other pin. For example, to see if the ADC is functional, outputting a DAC signal on the other pin provides a quick test of the function. A very simple application, RemoteIOHost was created to speed the development process. It provides a one-click ability to reconfigure each pin of the device, and view the incoming data messages. In the same way servicing devices in the field would be simple by using the device for self-testing.

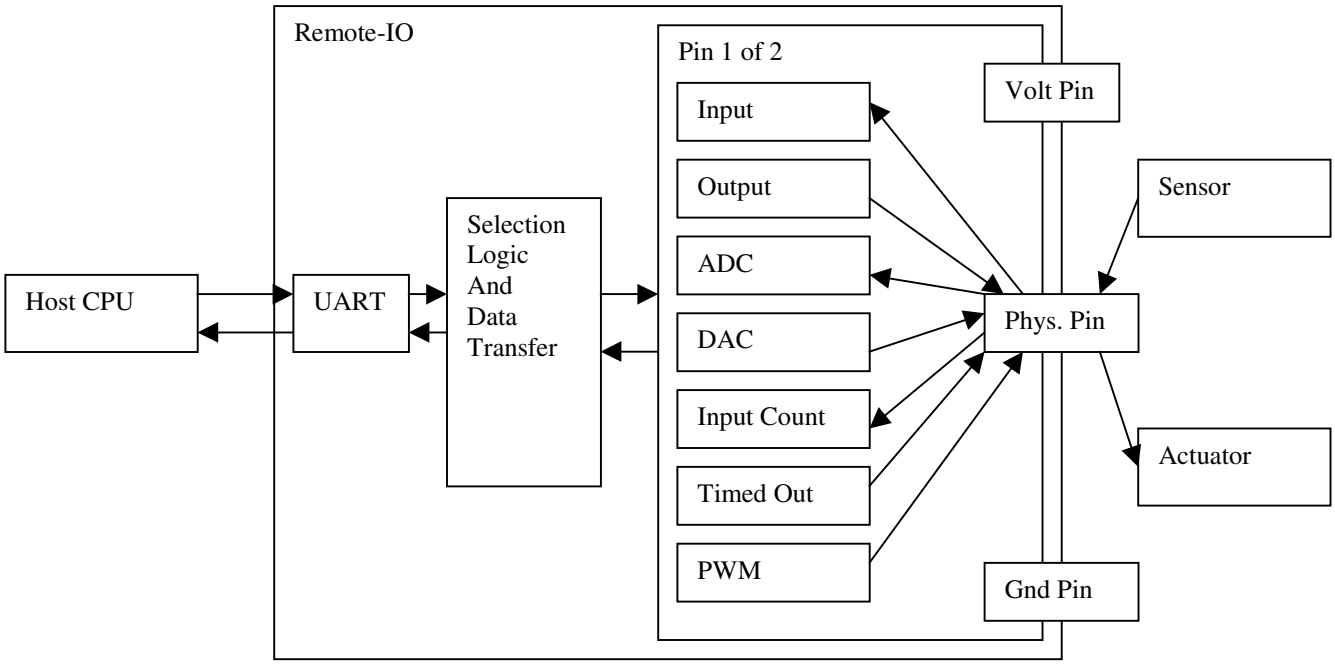
For a distributed network of devices, each device would be coded by a unique address or identification number. The microcontroller provides for EEPROM emulation in the Flash. Initialization of a device would consist of writing an address to each device so it could be accessed on the shared network. Each device held in stock would then be identical. An address would be applied when it was initialized, and its pins' functions could set. The function choices for the pins could also be set in the EEPROM. In the field, devices could be reconfigured as the need occurred. For troubleshooting machine problems, unused pins could be configured for test purposes, or sensor inputs could be simulated by using other Remote-IO device pins to provide stimulation. The flexibility of each device eliminates the need for special test devices.

In summary, a rich selection of input/output functionality is provided in a simple package in a user-friendly way due to the underlying flexibility of the device's hardware. Not only is the end user provided with an easy to use and service device, but the development is quick and efficient as well.

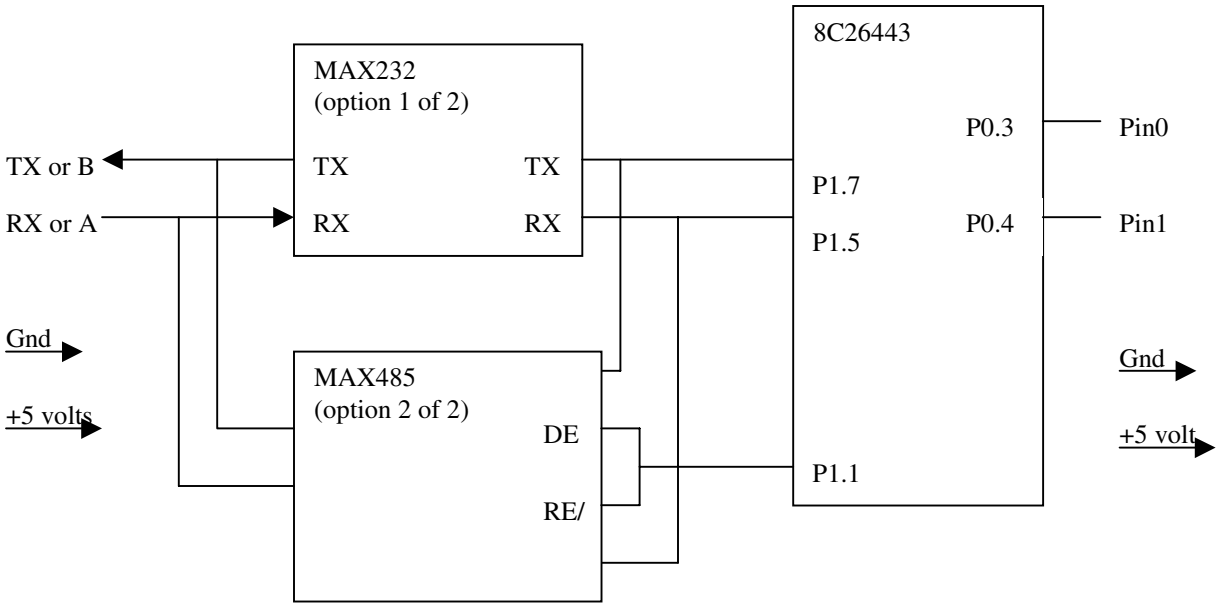




Block Diagram



Schematic



VCC: MAX232.16, MAX485.8, 8C26443.28
GND: MAX232.15, MAX485.5, 8C26443.14