GPIO Game Adapter

for MBC style single board computers, and other GPIO interfaces.

The Basics:

This interface is a simple to use, bare bones game port break out, with a built in YL-44 clone passive buzzer module and 555 interval timer. This can be used in design situations where only very basic IO is available. The simple design lends itself to certain limitations. However, it is quite sufficient enough to create simple and even more complex game's even under very archaic settings. For instance, with the use of this interface and 8 bit Pascal/MT+ we were able to accomplish a fully functional Tetromino Puzzle game, including sound effects and joystick support running on CP/M 2.2 without the use of any interrupts.

The Joystick interface:

The joystick uses 5 signal lines, all switched through a Schmidt Trigger to clean up any switch bounce off of even the lowest quality joystick. When using the reference source found linked later in this documentation, it is connected to pin's 3-7 on port A of the GPIO. This is to make reading the entire port as a specific value, very simple. Though other techniques can be employed if only one 8 bit interface is available, for instance bit shifting can be used to read each bit location to test for values.

The sound interface:

A single signal is needed to drive the buzzer, simply switching it on and off at speed can accomplish this, the reference source uses Pin 8 of Port B; and employs a technique of counting no operation instructions for a period of time to create a square wave as well as duration. This is obviously very CPU dependent but for short beeps and sound effects it is more than sufficient.

The 555 Reference Timer:

The timer uses the remaining inverter on the Schmidt trigger to provide a low triggering pulse to the 555, via a high signal being sent to the adapter interface. On the reference source this is on pin 7 of Port B of the GPIO. Simply sending a High signal starts the 555 timer, which is designed to run for 11ms though can be configured to other values by changing the resistor, capacitor combination (see the schematics for details). You can probe the status of the timer by checking the EXP signal from the interface, while the timer is running the status of the signal is HIGH, when the timer has expired it is LOW.

Using the timer in the context of the game can be simply done by intermittently checking the status of the expired signal and then augmenting a single byte to track the number of 11ms segments by 11. On the offset this seems like a fairly inefficient way of determining the passage of time but considering the only other option is counting instructions and cycles to determine exact timing; this may be considerably easier.

The timer's resolution is not fast enough to be useful for sound, graphics hardware or serial timing, but is definitely good enough to handle loop flow control timing within the context of a simple game loop, or even basic multi-tasking through time sharing.

The Build:

The PCB is a simple 2 layer design with two common planes, VCC is on the entire topside of the interface, while the ground plane is on the bottom. The BOM specifies Electrolytic capacitors, however ceramic of the same value will absolutely work. If a polarized cap is used be careful to not install them in the incorrect orientation. 1% tolerance resistors were used and the accuracy of the timer circuit was found to be very close. These types of compactor circuits tend to have very specific passive requirements in order to get desired results; so it's fairly reasonable to assume that if you use lower quality gates and passives, your timer may not be as accurate as you like. Provided is a Gerber, as well as a schematic and BOM if you wish to reproduce this project.

Reference source code for the interface:

Joystick – https://github.com/lindoran/minp
Sound – https://github.com/lindoran/MBC-GPIO-Sound
Time -- https://github.com/lindoran/MBC---Time

License: