

# Colecovision Multicart Manual

SPARKLETRON

October 27, 2024

Jay Convertino

# Contents

<b>1</b>	<b>Usage</b>	<b>2</b>
1.1	Introduction . . . . .	2
1.2	Dependencies . . . . .	2
1.3	Building . . . . .	2
1.3.1	Colecovision ROM . . . . .	2
1.3.2	Microcontroller Code . . . . .	2
1.3.3	PCB . . . . .	2
1.3.4	3D printed Case . . . . .	2
1.4	Directory Guide . . . . .	2
<b>2</b>	<b>Architecture</b>	<b>4</b>
2.0.1	General Description . . . . .	4
2.0.2	Excuses . . . . .	4
<b>3</b>	<b>Code Highlights</b>	<b>5</b>
3.1	Colecovision Multicart . . . . .	5
3.2	Python Generators . . . . .	5
3.2.1	ROM Header Generator . . . . .	5
3.2.2	ROM File Generator . . . . .	5

# 1 Usage

## 1.1 Introduction

This manual describes how to use the Multicart project and its architecture. This cart was design to be cheap, and easy to use. By using a very basic microcontroller and some inline assembly a reliable multicart has been created. This cart allows you to select between 15 different ROMS. These 15 ROMS can be any colecovision, or colecovision Super Game Module game you wish. Only caveat is it has to be 32 KiB or less. This cart does not support MEGA roms.

## 1.2 Dependecies

COMING SOON(TM)

## 1.3 Building

COMING SOON(TM)

### 1.3.1 Colecovision ROM

COMING SOON(TM)

### 1.3.2 Microcontroller Code

COMING SOON(TM)

### 1.3.3 PCB

COMING SOON(TM)

### 1.3.4 3D printed Case

COMING SOON(TM)

## 1.4 Directory Guide

```
docs/
├── datasheets
├── manual
├── schematic/
│   ├── gerber
│   └── step
├── src/
│   ├── pic_coleco_addr_sel
│   └── firmware
```

The above listing shows the important directories for this project. The list below describes each section.

1. **docs** Contains all documentation related to this project.
  - **datasheets** Contains all IC datasheets used for the hardware.
  - **manual** Contains user manual and wiki that are generated from the same latex source.
2. **schematic** Contains KiCAD schematic for the project, currently version 7.x of KiCAD.
  - **gerber** Export of schematic gerbers are placed here.
  - **step** Export of 3D step model here for case fitment.
3. **src** Contains all source code related to the project.
  - **pic\_coleco\_addr\_sel** pic16 source code and build system, uses xc8 compiler. Any version over 2.X should work fine.
  - **firmware** Colecovision development kit that uses SDCC. The multicart app will generate the project and build rom file.

## 2 Architecture

### 2.0.1 General Description

The overall architecture is fairly simple and has the following steps.

1. Boot colecovision in standard banner mode. Display multicart title.
2. Display selection screen in TMS text mode.
3. Highlight current user selection, defaulting at the 1st entry.
4. Once fire is pressed, the highlighted entry is selected for bank switching.
5. Colecovision code will load a new banner telling the user to reset the console.
6. Colecovision will spam the address  $E001 + \text{entry}(0 \text{ to } 14)$  4 times to activate PIC.
7. When  $E000n$  enable line goes low and the PIC catches it, the PIC will then set its output address lines to the rom to the input address lines.
8. User will reset console, console will now read the bank switched ROM from multicart.

### 2.0.2 Excuses

One of the major items you may notice is the need for the user to reset the console. This is due to my design, being a pain to time correctly. By making the user reset the console, which is much slower than the console, the colecovision and the PIC can be synched easily. Yes that PIC is the reason for this. The PIC at 20 MHz takes 200ns to execute each instruction. Currently my algorithm (see ??) will take 1000ns to sample, test the sample and then decide if  $E000n$  is active. The  $E000n$  pulse was measured to only be 620ns. Since the PIC is simply in a loop sampling, this means the pulse can be missed by the microcontroller. Easy way to fix this is to spam a few reads, in this case four, from the Colecovision Multicart program.

Now the PIC has to set the address. This of course is fairly quick, sometimes too quick and results in corruption. Mostly with Atari games. This resulted in the idea of using a halt instead of reset. Basically the Colecovision code spams the read four times, then goes to a halt. The PIC picks up the read, captures the address and then waits for 1 second. After than second it bank switches the ROM. Since the console is halted this presents no issue and the user can now reset the console to start their game.

Getting the timing between the two perfect could be achieved, heck others have done it. Frankly, this was the easy way out for now. Maybe in the future I'll revisit it. Though the goal is small and cheap. Three IC's is fairly small for the multicart, and as it is now works perfectly for my usecase.

## **3 Code Highlights**

### **3.1 Colecovision Multicart**

### **3.2 Python Generators**

#### **3.2.1 ROM Header Generator**

#### **3.2.2 ROM File Generator**