I^2C Analysis

Nickolas Gallegos

2023-04-24

Introduction

This document will be used to analyze capture data from the I^2C module for the *Wagner* Project.

The data is exported from the Saleae Logic Analyzer application and stored in the "Captures" folder of the R Project.

Data

The file is *i2c_2peripheral.csv* and contains a logic capture of interactions between one controller device (Raspberry Pi Pico) and two peripheral devices (Adafruit ItsyBitsy and Adafruit Trinket M0).

Read in Data

```
file <- "../Captures/i2c_2peripheral.csv"
data1 <- read.csv(file)

#View(data1)

head(data1)
```

```
##            name    type start_time  duration   ack address  read data
## 1           I2C   start   3.795227 4.000e-08            NA
## 2 I2C_STUSB4500    ping   3.795227 7.584e-05             8
## 3           I2C address   3.795244 4.472e-05 false       8 false
## 4           I2C    stop   3.795303 4.000e-08            NA
## 5           I2C   start   3.795423 4.000e-08            NA
## 6 I2C_STUSB4500    ping   3.795423 6.792e-05             9
##   description count action
## 1               NA     NA
## 2               NA     NA
## 3               NA     NA
## 4               NA     NA
## 5               NA     NA
## 6               NA     NA
```

```
tail(data1)
```

```
##     name type start_time  duration   ack address read data description count
## 571  I2C data   4.519317 2.920e-05  true      NA      0xFF                NA
## 572  I2C data   4.519351 2.920e-05  true      NA      0xFF                NA
## 573  I2C data   4.519385 2.920e-05  true      NA      0xFF                NA
## 574  I2C data   4.519419 2.920e-05  true      NA      0xFF                NA
## 575  I2C data   4.519454 2.864e-05 false      NA      0xFF                NA
## 576  I2C stop   4.519489 4.000e-08            NA                          NA
##     action
## 571     NA
## 572     NA
## 573     NA
## 574     NA
## 575     NA
## 576     NA
```

```
colnames(data1)
```

```
##  [1] "name"        "type"        "start_time"  "duration"    "ack"
##  [6] "address"     "read"        "data"        "description" "count"
## [11] "action"
```

Clean Data and create subsets

The data has column names:
- name
- type
- start_time
- duration
- ack
- address
- read
- data
- description
- count
- action

The important columns will be the *type*, *start_time*, *duration*, *address*, *read*, *data*

Get rid of the "I2C_STUSB4500" information (not certain what this is at the moment).

```
data1 <- data1[(data1[,"name"] != "I2C_STUSB4500"),]
```

Only keep the "important" columns.

```
cols <- c("type", "start_time", "duration", "address", "read", "data")
data1_sub1 <- data1[,cols]
#View(data1_sub1)
```

Now create subsets where the data comes from specific devices.

```r
address_type = data1_sub1[,"type"] == "address"
head(data1_sub1[address_type,])
```

```
##       type start_time  duration address  read data
## 3  address   3.795244 4.472e-05       8 false
## 7  address   3.795432 5.140e-05       9 false
## 11 address   3.795532 3.540e-05      10 false
## 15 address   3.796626 3.572e-05      11 false
## 19 address   3.796695 3.540e-05      12 false
## 23 address   3.797761 3.540e-05      13 false
```

Creating a function to convert hex data to ASCII (prints ASCII until it sees $0xFF$)

```r
convert_to_ascii <- function(data_vec)
{
  result <- c()
  for (data in data_vec) {
    hex <- as.hexmode(data)
    if (hex != 0xFF){
      ch <- rawToChar( as.raw( hex ) )
      result <- append(result,ch)
    }
  }
  return(result)
}

#vec <- c("0x3A","0x42","0xFF")

#convert_to_ascii(vec)
```

```r
data_type <- data1_sub1[,"type"] == "data"

d <- data1_sub1[data_type,"data"]
dat <- convert_to_ascii(d)


paste(dat,collapse = "")
```

```
## [1] "Hello from the ItsyBitsy!Hello from the Trinket M0!"
```