

PlayNPort: A Portable Wireless Music Player and Text Reader System

Lakhan S. Kamireddy¹, D. Thakkar¹ and Lakhan Saiteja K.²

¹Department of Electrical, Computer and Energy Engineering
University of Colorado Boulder

²Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology Kharagpur

IEEE International Conference on Consumer Electronics
January 11-13, 2019
Las Vegas, NV

Table of contents

- 1 Board Design
- 2 Communication
- 3 Firmware Design
- 4 Challenges
- 5 Conclusions
- 6 Acknowledgements
- 7 References

Motivation for this work

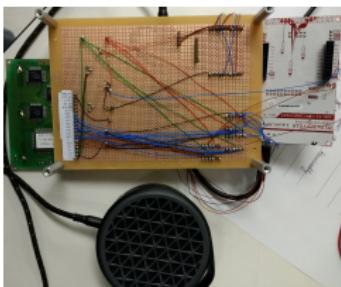


Figure: Board backside

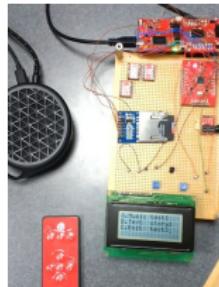


Figure: Board frontside

- Bare metal firmware design.
- Consumer-centric product design with low cost electronics.
- Offer a rich user experience, yet with minimal configuration.
- Identify design challenges, and document solutions for use.
- Offer an open, complete and ready-to-use source code for educational purpose.

System Block Diagram

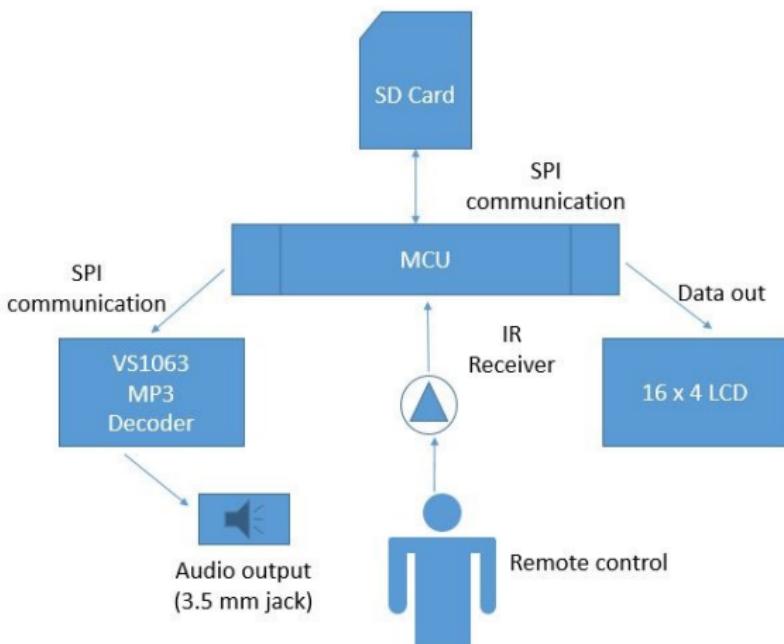


Figure: System Block Diagram

Board Design

- **LCD Interface:** Character LCD for menu and messages
- **SD Card Interface:** With SD card breakout board, MISO, MOSI and SCLK to eUSCI B2 module (SPI Mode) on MSP432
- **IR Module:** Wireless remote control, play/pause, prev/nxt, vol up/dwn for music, exit to menu
- **MP3 Decoder & 3.5mm jack:** VS1063 Decoder/Encoder IC for mp3/wav formats, TRRS Jack

Wired Communication

Serial Peripheral Interconnect (SPI)

- Master/Slave setup, runs in Full Duplex, speeds upto 20 Mbps
- Configured TI MSP432 as Master, SD Card and VS1063 IC as slaves
- Data Transfer: TXBUF → TXS
- SOMI → RXS (\uparrow opp clk edge)
- RXS → RXBUF. Reads when *RXIFG* flag is set

Bit Rate Control:

$$f_{uclk} = f_{BitClock} = f_{BRCLK}/UCBRx$$

Wireless Communication

NEC Protocol for Wireless IR

- Standard protocol using pulse distance encoding of the bits
- Logical 1 → 2.25 ms, logical 0 → 1.125 ms, LSB First

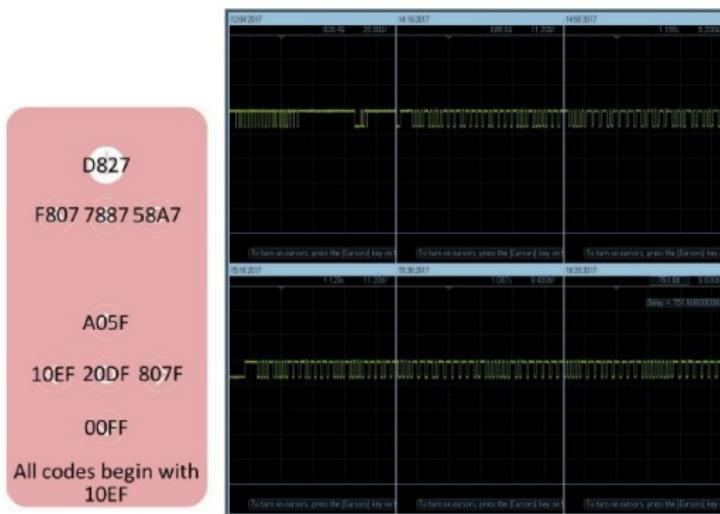


Figure: Capturing IR Signal: A, B, C, <, ^, >

Firmware Design

SPI Driver: Reading/Writing registers of eUSCI_B module **SDHC**
Background:

- SD Card responds to a set of predefined commands (SD Card Association)
- Six Registers: OCR, CID (Card Iden), CSD, RCA, DSR, SCR (Config)

SDC Initialization & Operation:

- Cmd/Resp: CMD0 → 0x01, CMD8 → 0x01, CMD55 → 0x00
- 010 → Data Accepted, 101 → CRC Error, 110 → Write error

Firmware Design

File System

- We used FAT File System on the SD Card, FAT32 for 4GB to 32GB
- Ported generic FAT file system, FatFs into our application, implemented MSP432 specific methods

Audio Decoder Drivers

- To enable VS1063 for audio decoding
- Initialize SPI & GPIO pins, Read/Write SDI (Serial Data Interface), SCI (Serial Command Interface)

Firmware Design

IR Transmission, Reception and Decoding

- *IR Transmission:* Pulse length encoding, start bit (9ms) leading pulse burst, 4.5ms space, 8-bit address (recv device), its 8-bit logical inverse, 8-bit command, its 8-bit logical inverse
- *IR Decoding Logic:* We used MSP432 timer module to decode bits

LCD Driver

- Initialize, read data, poll LCD busy flag, put command/data on LCD's data lines, etc.

Challenges

- Issue Mounting FAT File System
- Software initialization of VS1063 Audio Decoder
- VS1063 not responding to SDI data write
- Was that music not playing at all or was it playing at a really fast speed (that we can't catch up with) ?

Conclusions and Future Work

- We developed both the hardware and software of system in its entirety with \$70 electronics components
- By writing bare-metal firmware code, we achieved full control over the system
- Able to provide rich user experience through options like play/pause, prev/next song, scroll down while reading text
- Offer source code for educational purposes

Acknowledgements

- We would firstly like to thank the audience for their interest.
- We would also like to thank the organizers for giving us an opportunity to present here.
- Last, but not least we thank Prof. Linden McClure, in whose lab we worked out most parts of this work.

References |

-  *Dot Matrix Liquid Crystal Display Controller Datasheet*, Hitachi, pp 167-226, 2010.
-  *Using MSP432 serial modules*, Online, 2015, Accessed from <http://www.samlewis.me/2015/05/using-msp432-eUSCI/>.
-  *IR Receiver Modules for Remote Control Systems*, Vishay Semiconductors, pp 1-7, 2008.
-  *VS1063a Encoder and Audio Coded circuit*, VLSI Solution, pp 1-93, v 1.31, 2017.
-  *Serial Peripheral Interface (SPI) User Guide*, Texas Instruments, pp 1-51, 2012.

References II

-  Bai Ying,
Microcontroller Engineering with MSP432: Fundamentals and Applications,
CRC Press, 2016.
-  *Enhanced Universal Serial Communication Interface (eUSCI) – SPI Mode,*
SLAU424F, 2012, Texas Instruments.
-  *NEC IR Remote Control Interface,*
Altium Techdocs, Online, 2017
-  *Sandisk microSDHC OEM Product Manual*, pp 1-25,
Western Digital Inc., 2016.
-  Elm, Chang, *How to use MMC/SSD*, 2010,
Accessed from http://elm-chan.org/docs/mmc/mmc_e.html.

References III

-  Elm, Chang, *FAT Filesystem*, 2010,
Accessed from http://elm-chan.org/docs/fat_e.html.
-  *VS1063 AppNote: Playback And Recording*,
VLSI Solution, pp 1-7, v 1.10, 2016.
-  *VS1063a Patches: VLSI Solution Audio Decoder/Encoder*,
VLSI Solution, pp 1-16, v 2.01, 2017.
-  Stefan Schauer, Christian Speck,
*App Note: Interfacing the MSP430 With MMC/SD Flash
Memory Cards*,
Texas Instruments, pp 1-5, 2008.

References IV



K Lakhan Shiva,

Wireless Music Player and Text Reader using TI MSP432,

Accessed from

<https://github.com/lakhanshiva/SDCardController>