Department of Computer Science and Engineering  
The University of Texas at Arlington

Team Sketchers

Sidewalk Sketcher

Team Members:   
*Jesus Parra  
Pranil Maharjan  
Sabin Bajracharya*

*Nischal Pandey  
Lam Pham*

Late Updated: 5 February 2015 @ 10:12:00 PM

Table of Contents

[1. Introduction 4](#_Toc411021423)

[2. Architecture Overview 4](#_Toc411021424)

[3. System Hardware Description 4](#_Toc411021425)

[3.1 Raspberry Pi Model B+ 4](#_Toc411021426)

[3.2 2GB Micro SD Card 5](#_Toc411021427)

[3.3 Raspberry Pi Camera 6](#_Toc411021428)

[3.4 USB-Serial Cable 6](#_Toc411021429)

[3.5 Roomba® iCreate 2 6](#_Toc411021430)

[3.6 Stepper Motor 6](#_Toc411021431)

[3.7 LED’s 6](#_Toc411021432)

[3.8 Broom Sticks 6](#_Toc411021433)

[3.9 Swim Noodle 6](#_Toc411021434)

[3.10 Speakers 6](#_Toc411021435)

[3.11 Plywood 6](#_Toc411021436)

[4. Software Input Layer 6](#_Toc411021437)

[4.1 Image Reader 6](#_Toc411021438)

[4.2 Converter System 6](#_Toc411021439)

[4.3 Transfer Data 6](#_Toc411021440)

[5. Software Output Layer 7](#_Toc411021441)

[5.1 Data Packager 7](#_Toc411021442)

[5.2 File Generator 7](#_Toc411021443)

[5.3 File transfer 7](#_Toc411021444)

[6. Software Processing Layer 8](#_Toc411021445)

[6.1 Image Processing 8](#_Toc411021446)

[6.2 Information Processing 8](#_Toc411021447)

[7. User Interface Layer 9](#_Toc411021448)

[7.1 File Browser Subsystem 9](#_Toc411021449)

[7.2 Cropping Subsystem 9](#_Toc411021450)

[7.3 Resize Subsystem 9](#_Toc411021451)

[7.4 Color Selector Subsystem 9](#_Toc411021452)

[8. Data Storage Layer 9](#_Toc411021453)

[8.1 Image Repository 9](#_Toc411021454)

[8.2 File Repository 9](#_Toc411021455)

[8.3 Database Management 9](#_Toc411021456)

[9. Hardware Input Layer 9](#_Toc411021457)

[9.1 File Reader Subsystem 9](#_Toc411021458)

[9.2 Synchronization Input Subsystem 10](#_Toc411021459)

[9.3 Camera Processing 10](#_Toc411021460)

[10. Hardware Output Layer 11](#_Toc411021461)

[10.1 Alarm Subsystem 11](#_Toc411021462)

[10.2 Output De-multiplexer 11](#_Toc411021463)

[11. Hardware Processing Layer 11](#_Toc411021464)

[11.1 Hardware Input Driver Subsystem 11](#_Toc411021465)

[11.2 Hardware Output Driver Subsystem 11](#_Toc411021466)

[11.3 Synchronization Subsystem 11](#_Toc411021467)

[11.4 Sketcher Processing Subsystem 11](#_Toc411021468)

[11.5 Position Processing Subsystem 11](#_Toc411021469)

[12. Sketch Layer 11](#_Toc411021470)

[12.1 Sketcher Synchronization Subsystem 12](#_Toc411021471)

[12.2 Depletion Subsystem 12](#_Toc411021472)

[13. Motion Layer 13](#_Toc411021473)

[13.1 Motion Synchronization Subsystem 13](#_Toc411021474)

[13.2 Motion-Driver Subsystem 13](#_Toc411021475)

[14. Quality Assurance 14](#_Toc411021476)

[15. Requirements Mapping 14](#_Toc411021477)

[16. Acceptance Criteria 15](#_Toc411021478)

[17. Appendix 15](#_Toc411021479)

# Introduction

This document provides a detail design specification of the robot Sidewalk Sketcher. It will explain the project over view as well as elaborate the hardware and software components used in the robot. This document will describe the architecture over view of the robot and give an insight description of layers used. In addition it will provide knowledge about Sidewalk Sketchers’ subsystems and their modules. Next the document will provide an explanation of how each layer, subsystem and modules communicate through dataflow diagram. A table of requirement mapping is provided to show how the requirements are accomplished by the robot and finally, an over view of acceptance criteria is provided for the quality assurance.

# Architecture Overview

# System Hardware Description

This section describes the hardware components that will make up the Sidewalk Sketcher. The hardware information provided here covers the quantity required, manufacturer specifications, its intended role in the system, a brief description of how it operates, and other components that will interface with the hardware.

## Raspberry Pi Model B+



**Figure 3-1** Raspberry Pi Model B+

### Quantity: The Sidewalk Sketcher will require one Raspberry Pi.

### Purpose: The Raspberry Pi will be the main processor of the Sidewalk Sketcher. This device will program the iCreate, control the chalk flow, and just overall control the physical component of the Sidewalk Sketcher. It’s 40 GPIO pins will be used to retrieve and send information to the iCreate and a stepper motor controlling the chalk unit. This device will also be in charge of receiving the output provided by the user interface and using this information to sketch an image.

### Specifications:

|  |  |
| --- | --- |
| Model | Raspberry Pi B+ |
| SoC | Broadcom BCM2835 |
| CPU | 700 MHz ARM1176JZF-S core |
| GPU | 25 MHz Broadcomm IV |
| RAM | 512MB |
| USB | 4 2.0 Onboard USB ports |
| Video Out | HDMI uo to 1920x1200 resolution, PAL, NTSC |
| Audio Out | 3.5mm Jack, HDMI |
| Storage | Onboard SD/MM/SDIO card slot |
| Network | 10/100mbps Ethernet, RJ-45 |
| Peripherals | GPIO, UART, SPI, IIC +3.3, +5.0V |
| Power | 700 mA, 5V via MicroUSB or GPIO header |
| OS | Raspbian |

### Interfaces: Since the Raspberry Pi is the central unit of control, it will interface with the iCreate, a stepper motor, LEDs and speakers.

## 2GB Micro SD Card



**Figure 3-2** 8GB Micro SD card

### Quantity: The Sidewalk Sketcher will require one 2GB micro SDHC memory card.

### Purpose: The operating system running on the Raspberry Pi will be stored on the memory card. Most of the system data will also be stored on the SD card.

### Specifications:

|  |  |
| --- | --- |
| Model | SanDisk |
| Capacity | 2GB |
| Class | 4 |
| Transfer Rate | 4Mbps |

### Interfaces: The SD card will interface with the Raspberry Pi via its SD slot.

## Raspberry Pi Camera

## USB-Serial Cable

## Roomba® iCreate 2

## Stepper Motor

## LED’s

## Broom Sticks

## Swim Noodle

## Speakers

## Plywood

# Software Input Layer

The purpose of the Software Input Layer is to accept input from the User Interface and output the image in way the system can manipulate. This layer is responsible for reading the image file, converting the image to an appropriate data file, and outputting the data to the Software Processing Layer for final processing.

## Image Reader

## Converter System

## Transfer Data

# Software Output Layer

The purpose of the Software Output Layer is to provide processed information or requested data to the hardware component in a format that it will be able to process. The output of the software component for the Sidewalk Sketcher will be handled in this layer and this will ultimately serve as the input for the hardware component.

## Data Packager

## File Generator

## File transfer

# Software Processing Layer

The purpose of the Software Processing Layer is to handle all the software input data, process that data and then send it to the output layer. Also, the processing layer should retrieve and save files in the data storage layer. This layer will handle all of the processing involved in the back end logic that the user will request through the User Interface Layer.

## Image Processing

## Information Processing

# User Interface Layer

The User Interface Layer allows the user to select an image that will be used by Sidewalk Sketcher to draw an image. It will also allow user to crop and resize an image to provide flexibility to users to sketch desired size and portion of selected image. The interface will provide an option to view the chalk color that will be used when the image is actually being sketched. This layer however, will not insure that the image looks in real life like it will look in the user interface because the actual color will depend on what colors are actually loaded into the Sidewalk Sketcher.

## File Browser Subsystem

## Cropping Subsystem

## Resize Subsystem

## Color Selector Subsystem

# Data Storage Layer

The Data Storage Layer contains all the subsystems that manages and holds a repository of all the image and data files saved by or accessed by the application. These images and data files may be requested from the Database Manager Subsystem for Image Processing in the Software Processing Layer.

## Image Repository

## File Repository

## Database Management

# Hardware Input Layer

This layer is responsible for reading input from Software Output Layer, Sketch layer and Motion layer, packaging all these inputs and sending it to the hardware processing layer. This layer includes the File reader subsystem, Sensor Reader subcomponent, Transfer Data, Camera Processing and Synchronization Input Subsystem. Input from Power Button and Start Button is also read in this layer.

## File Reader Subsystem

## Synchronization Input Subsystem

## Camera Processing

# Hardware Output Layer

All the hardware outputs in the form of light and sound are taken care of in this layer. It includes alarm subsystem, output De-multiplexer and Light subsystem. This layer depends on the output from hardware output driver subsystem of Hardware processing layer.

## Alarm Subsystem

## Output De-multiplexer

# Hardware Processing Layer

The purpose of the Hardware Processing Layer is to analyze and process data as well communicate with the remainder of the hardware layers: Hardware Input Layer, Hardware Output Layer, Sketch layer and the Motion Layer. This layer is the central processing unit of the hardware. Essentially this is the microcontroller in the hardware that controls all of the motions including the Hardware Input Layer, the Hardware Output Layer, the Sketching Layer and the Motion Layer. The sections below provide a detailed description of this layer and its subcomponents – Hardware Input Driver Subsystem, Hardware Output Driver Subsystem, Synchronization Subsystem, Position Processing Subsystem, and the Motion Subsystem.

## Hardware Input Driver Subsystem

\*\*\*Picture of whole Subsystem split into modules\*\*\*\*

### Receive Raw Data

#### Description:

##### This module will receive raw data from the hardware input layer. This data will contain a python list structure that will contain each line of the file loaded from the user interface. Each element of the list will contain a line in the file which will represent instructions for the Sidewalk Sketcher to follow. This module will also include the current position of the robot. In this module the system will store the set instructions and location and organize it in a way that can be passed to the Synchronization subsystem.

#### Interfaces:

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Parameter | Data Returned |
| getRawInput | Receive the input from the Hardware Input Layer | None | Python list of 2 elements, one with another list of instructions and the other of position coordinates |

#### External Data Dependencies:

##### None

#### Internal Data Dependencies:

##### Python list containing 2 elements

#### Pseudo Code:

##### 

## Hardware Output Driver Subsystem

## Synchronization Subsystem

## Sketcher Processing Subsystem

## Position Processing Subsystem

# Sketch Layer

The purpose of the Sketch Layer is to analyze all of the actions that will be performed by the sketching device that will drag the chalk, pick up the chalk when it’s not drawing, and send the update to the Hardware Processing Layer when the chalk nears depletion This layer will communicate directly to the Hardware Processing Layer and the Hardware Input Layer. With the Hardware Processing Layer, it will commute to let the processor know whether or not the chalk is near depletion along with whether the device should be in its writing state or in its floating state (floating meaning it is picked up because it is passing an area that there is no lines to be drawn) This layer will communicate with the Hardware Input Layer because it will have to send senor information to make sure the chalk is connecting to the ground when it is supposed to be sketching the image.

## Sketcher Synchronization Subsystem

## Depletion Subsystem

# Motion Layer

The purpose of the Motion Layer is to analyze the mechanical motion that will perform from the Sidewalk Sketcher and to ensure that the device is on the correct path. This layer will communicate directly to the Hardware Processing Layer and the Hardware Input Layer. With the Hardware Processing Layer which will commute to let the processor know where in the robot is at any given position so that the processing unit can determine where to go next and let this layer know the updated information. This layer will communicate with the Hardware Input Layer because it will have to send the positioning data collected from the data and sync this input to know its absolute position relative to the marker devices used for positioning.

## Motion Synchronization Subsystem

## Motion-Driver Subsystem

# Quality Assurance

# Requirements Mapping

This section provides an overview of the key system requirements and how each layer provides its functionality to meet the requirement through a table. Note that the layers present in the table are only layers that complete key requirements. Some layers that are required to run the system are not present in this table.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement Number | Requirement Name | Hardware Input Layer | Hardware Output Layer | Sketch Layer | Motion Layer | UI Layer | Hardware Processing Layer |
| 3.1 | Sketch Image |  |  | x |  |  |  |
| 3.2 | Sidewalk Sketcher Multicolor |  |  |  |  | x |  |
| 3.7 | Sketch Dimensions |  |  |  |  | x |  |
| 3.8 | Chalk Switch |  |  |  |  |  | x |
| 8.1 | Image Loading |  |  |  |  |  | x |
| 8.5 | Push Chalk |  |  |  |  |  | x |
| 8.6 | Finished Image |  | x |  |  |  |  |
| 8.7 | Depletion Sensing | x |  |  |  |  |  |
| 8.8 | Chalk Reload |  | x |  |  |  |  |
| 8.9 | Positioning |  |  |  | x |  |  |

# Acceptance Criteria

# Appendix