

Advanced クラスマイコンカーラリーの研究

研究者：小寺 晃士朗・天木 開士

1 はじめに

私たちが一年生の時から行ってきたマイコンカーラリーを通して、Advanced 部門・Camera 部門の車体の改善・改良を行った。

2 研究の内容

Advanced 部門では今まで使用していた車体の機構をアッカーマン機構に改良を行った。

Camera 部門ではルール変更に従って以前まで使用していた車体の改良を行った。

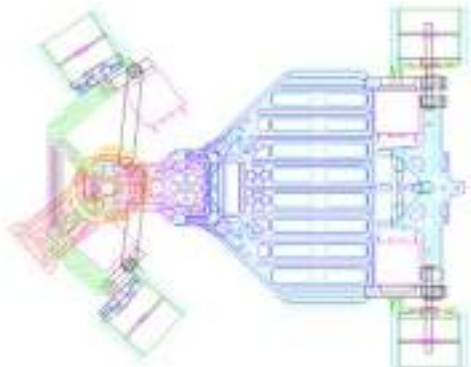
3 研究過程

- 3～5月 : 機構の理解と図面製作
- 5、6月 : 図面をもとに削り出し組み立て
- 7月 : 県大会に向けた調整
- 8月 : JMCR 岐阜県大会
- 9、10月 : 東海大会に向けての改善
- 11月 : JMCR 東海地区大会
- 12、1月 : 資料の整理と発表

4 研究の成果

Advanced 部門

- (1) Auto Cad・Cut 2D を使用しての製作
4月から Auto Cad を使用し、設計を行ってきた図面から Cut 2D を使用してジュラルミン板やカーボン板などから削り出すことができた。



Auto Cad で製作した図面



Cut 2D を使用し具現化

(2) カーブ・車線変更での変化

アッカーマン機構車体の特徴としてカーブが曲がりやすいというものがある。今回製作を行ったアッカーマン機構の車体もカーブで滑らず、内側を走ることができた。

車線変更では今まで使用していた車体と比べ、浅い角度で曲がることができた。

クランクでは、今までの車体と比較しスムーズに曲がりきることができた。



(3) 大会結果

8月9日

小寺 晃士朗
県大会 9位
天木開士
県大会 6位

11月11日

小寺 晃士朗
東海大会 17位
天木開士
東海大会 5位

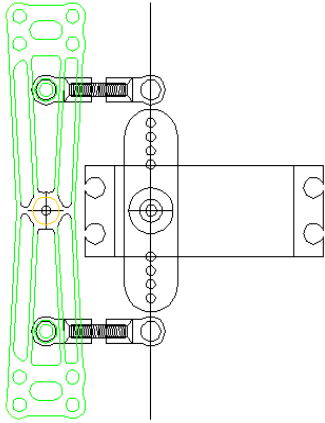
Camera 部門

- (1) ルールにあった今よりも大きい基板を搭載できる車体の制作

去年の東海大会では、使用していた基板がルール上のもではなかったため、優勝ができたものの全国大会に出場することができなかった。そのため、大会に正規出場ができるように、新しい基板を搭載できる車体を、設計・制作した。

(2) 棒を使用してカメラがサーボモーターと一緒に動く車体の制作

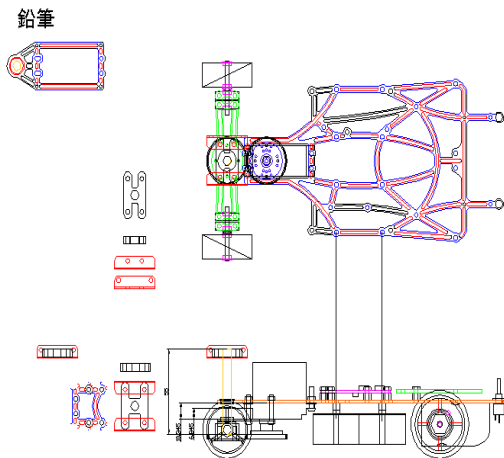
新しいルールでは、以前まで許可されていなかった「カメラが動く車体」の使用が許可されたため、新しいことに挑戦すべく、カメラがサーボモーターと一緒に動く車体の設計を行った。



作成した図面

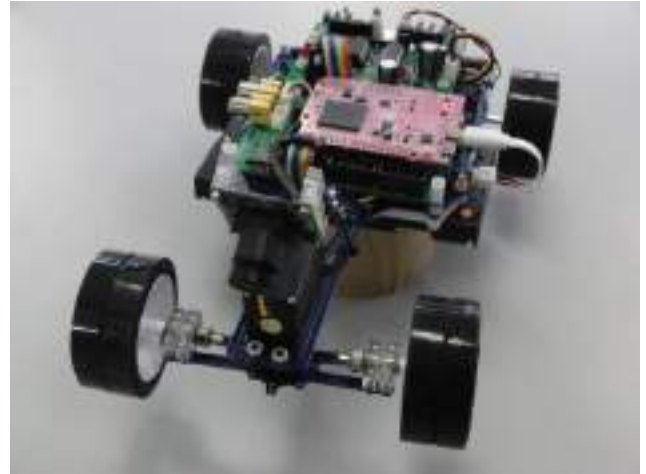
(3) シャフトを使用したサーボモーターと一緒に動く車体の制作

棒を使用した車は、却下されたので、次にシャフトを使用してカメラを動かせる車体の設計をしました。



作成した図面

(4) カメラが動く車体の基礎を生かして、カメラが動かない車体を制作
度重なるルール変更で、最初に設計・制作した車体も使用できないことになったため、カメラが動く車の基礎を生かしてカメラが動かない車の設計・制作を行いました。



最終的に完成した車体

5 まとめ

Advance 部門

今までの車体から大きく改良を行いアッカーマン機構へと変更を行った。
今まで使用してきたソフトなどを使用し、0から完成させることができた。県大会などでも成果を残すことができた

Camera 部門

新しいルールの決定によって、使用できなくなった車体を修正し、カメラが動く車も制作した。その後のルールの変更に合わせて、ルールに合った車体の基礎も定着させることができた

6 チームの感想

【小寺 晃士朗】

一から新たなものを作ることはわからないことも多く、時間もかかり大変なことも多かったが、車体を完成したときの達成感や県大会・東海大会などで成果を出せることに喜びを感じ取り組むことができた。また、新たなものを作る時は問題点に対する改善と根気強さが大切だと感じた。

【天木 開士】

車体を作るにあたって、失敗することやつまずく事もありましたが、作成した車体が動いた時の喜びは、格別でした。
結果は自分の思うようには出なかったですが、新しいことにチャレンジできた経験はとても貴重なものだと実感しました。

Basic class マイコンカーの製作

研究者 早野 晃生
坂 直郁
三宅 航生

1 はじめに

私たちが3年間学習してきたことを、生かすことができることに加え、新たな分野に踏み出すことで、新たな知識を蓄えるため課題研究を行った。

2 研究の内容

Basic クラスのマイコンカーを受け継ぎ、そのマイコンカーのプログラムや車体を改良し、より速くタイムを縮めることを目指す。また、大会において上位入賞、優勝を目標とし、研究に取り組む。

3 研究過程

- 4、5月 : 車体の整備
- 6、7月 : プログラム改良、車体改良
- 8月 : 岐阜県大会
- 9、10月 : プログラム改良
- 11月 : 東海大会
- 12、1月 : レポート作成、資料作成

4 研究の成果

(1) マイコンカーの車体について

先輩から受け継いだ車体を使用し、メンテナンス、改良をしていった。

Basic チームは全員、3DCAD を使用し各々タイヤを設計し、3D プリンタでタイヤを制作した。また、車体によって構造が少しずつ違ったり、サーボの値、モーターの強さも違っていたため一人一人がそれぞれ個性のある車体を改良した。

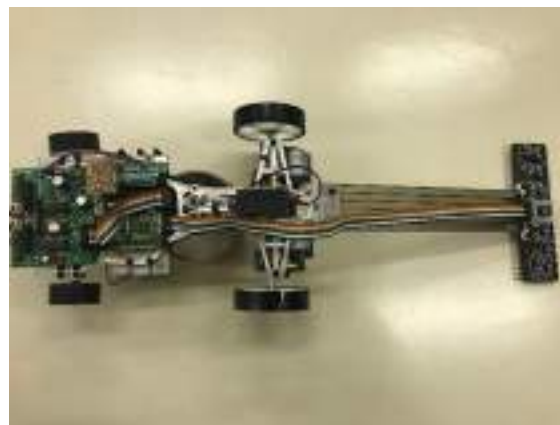


図1 マイコンカー

(2) 各大会成績

- 岐阜県大会 3位 三宅 航生
- 8位 坂 直郁
- 9位 早野 晃生

8月9日(金)

会場 : 大垣工業高等学校



図2 岐阜県大会

東海大会 22位 三宅 航生

11月10日(日)

会場 : 可児工業高等学校

岐阜県大会では全員完走し入賞を果たすことができた。しかし東海大会では、一人しか完走せず、順位も課題の多く残るものとなってしまった。



図3 東海大会

(3) プログラムについて

プログラムは元となったプログラムは同じですが、車体によって合うプログラムと合わないプログラムがあった。

今年度、元となったプログラムに追加したプログラムは大きく分けて2つある。

- ①自動停止プログラム
- ②EEP-ROM 基盤を使用したログを記録することができるプログラム

①について

自動停止プログラムは、センサがすべて反応している状態がプログラムで定めた秒数続いたらモーターの値を0、ハンドルの値も0に設定するもので、コースアウトした際に設定した秒数センサが反応したら停止させるプログラムです。

②について

EEP-ROM 基盤を使用したログを記録することができるプログラムは、EEP-ROM 基盤を使用し、車体をコースで走らせた際に、細かいパターンの変化やセンサの変化、さらにはモーターの値までを記録してパソコンに出力し、目測では確認できなかった細かい変化を見つけ、改良するためのプログラムです。

5 まとめ

Basic クラスマイコンカーの製作を通して、まず、全員が大会において一度は完走し、さらに入賞を果たすことができたことが成果として挙げられる。また、一人一人が車体のタイヤを制作し、車体のメンテナンス、さらにプログラム改良が一人でできるようになったことも挙げられる。しかしその反面、大会において完走しきれなかったことや、調整に時間がかかりすぎることなどの、課題も残る結果となった。

6 チームの感想

【早野 晃生】

失敗も成功も含めて、自分の力にすることができたのでこの力を将来に生かせるようにしておきたいです。

【坂 直郁】

様々なプログラムを扱ったことでプログラムに対する理解力を高めることができました。また、モノづくりの厳しさと大切さを学ぶことができました。

【三宅 航生】

初めてやることで始めは戸惑いましたが、自分のスキルが磨けたので良かったです。



ロボトレサの開発・改善

研究者：山岸 蓮弥、小池 正基
大橋 裕斗、桐山 大司

1 はじめに

情報技術部で取り組んできたロボトレサ競技に出場するための車体開発についてさらに研究し、より高性能な車体を作ることを目的として研究を行った。

2 ロボトレサ競技について

ロボトレサ競技は、黒い床に引かれた白いライン（1周 60m 以下）の周回コースをできるだけ速く走ることを競うものである。コースにはカーブやスラローム、クロスラインなどがあり、それらをうまく走行し完走しなければならない。また、曲率が変化するところにマーカーが設置されており、それにより直線やカーブの曲率を認識することができる。スタートやゴールにはそれぞれスタートマーカーとゴールマーカーがあり、それでスタートとゴールを認識し、走行を行う。大会では3分間に3回走らせることができ、その中でコースの形状に合わせて適切な走行ができることが推奨される。

コースの形状に合わせて走行するためには、区画ごとの長さやカーブの曲率などを知る必要がある。そこで1走目にエンコーダを用いて、マーカー毎の距離や曲率、カーブの半径など必要な情報を記録する。

2走目からは、1走目に記録した情報をもとに、直線や大きいカーブの場合は速度を上げ、カーブの直前にはブレーキをかける。カーブの角度によって細かく速度を設定することで、1走目よりも速く、効率の良い走行をさせることができる。

3 研究の内容

(1) 新車体の設計

モーターの制御を行う基板が過負荷により壊れてしまうことと、モーターに定格 6V 以上の電圧を加えるため、自作の基板を追加してモーター制御基板の増設を行った。



図1 追加した基板

カーボンでシャーシ(車体のフレーム)を切り出し、その上に基板やモーターを置く形で車体を製作した。シャーシを骨組みだけのもの(厚さ 1.5 mm)と、素材は同じだが厚さが薄いもので作り分けた(厚さ 1.2 mm)。



図2 厚さを変えた2種類のシャーシ

マイコン基板は VS-WRC103LV を、センサ基板は VS-IX010 を使用した。

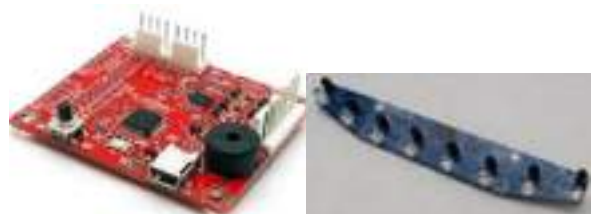


図3 マイコン基板とセンサ基板

(2) シリコンタイヤの製作

今まではシリコンシートを使ったタイヤを使っていたが、摩擦力と耐久性の向上を図り、シリコンタイヤを作ることにした。

シリコンタイヤは、液化シリコンに希釈材、硬化剤を混ぜ、型に流し込むことで製作する。



図4 製作中のタイヤ

シリコンタイヤを作る際に必要な金型は、機械科の先生にお願いし、旋盤で製作した。

シリコンタイヤを製作する際、いくつかの寸法でタイヤを作り、比較実験することにした。その中で最もよかったものが、横幅 15mm、厚さ 1mm、内径 26mm のものである。

(3)プログラムの改良

①ライン追従制御

ロボトレサは、ラインに追従するためのプログラムをPD制御で処理している。

センサで読んだ白線が車体の中心からどれだけ離れているか計算し、白線が中心に近ければ弱く、遠ければ強くモーターが回転し、左右の回転差でラインを追従する。

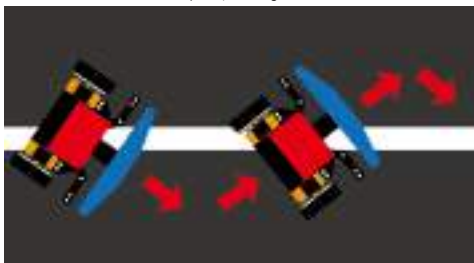


図5 P制御の動作

P制御だけでは、車体が中央に近づく勢いが強すぎて、車体が左右に揺れてしまう。

PD制御を使用することにより、ラインに近づく勢いを調整することで、ライン中央に沿って走ることが可能となった。

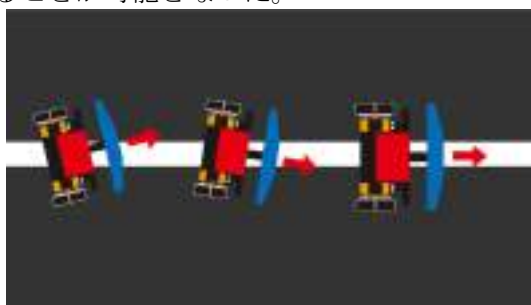


図6 PD制御の動作

②クロスライン処理、マーカーセンサ処理

コース上で線が直交する場所をクロスラインと呼び、マーカーとの誤検知をできるだけ減らすために専用の処理を改良した。また、競技規格でクロスラインは $90 \pm 5^\circ$ まで傾くことがある。そこで、マーカーセンサの読み取り制御を変更し、クロスラインが少し傾いていても、誤検知をしないようにした。

③角速度フィードバック制御

今までのライン追従制御では、トレサ前面の8連センサのみを使って制御していたが、そこにエンコーダを使った角速度フィードバックを追加した。左右の回転数を確認しながら制御を行えるため、全体的に走行を安定させることができた。

4 研究の成果

(1)新車体の設計

骨組みだけのものではパーツを切り分けて

無駄な部分を省くことで、シャーシを軽く作ることができた。素材の厚さが薄いものでも、正常に走行できた。

(2)シリコンタイヤの製作

シリコンタイヤの製作は難しく、さらに摩擦もシリコンシートに比べて発揮されなかったため、これからもシリコンシートタイヤを使うことにした。

(3)プログラムの改良

クロスライン処理での誤検知が減った。加減速処理をするときの速度の目標値をより細かく、正確に指定できるようになり、滑らかな加減速ができるようになった。角速度フィードバックにより、ライン追従性が上がった。

5 大会出場

2019年度マイクロマウス関西地区大会 (7/1)

ロボトレース競技 6位、8位、13位

第34回マイクロマウス中部地区初級者大会 (9/1)

ロボトレース競技 準優勝、3位

第40回全日本マイクロマウス大会 (12/1)

ロボトレース競技 22位、50位、51位、56位

6 チームの感想

【 山岸 蓮弥 】

今回の研究から、より良い車体やプログラムを作ることができたのでよかった。また、大会で他の高校の人や大学生からいろいろな刺激を受けた事がとても良い経験だった。

【 小池 正基 】

この研究を通して、より速いロボトレサを作ることができた。この研究を通して、金属加工や車体設計、プログラミングなど、幅広い技術を学ぶことができたので、大学でも生かしていきたい。

【 大橋 裕斗 】

課題研究を通して、ものづくりの大変さを知ることが出来た。1から車体を製作したことで設計すること、加工すること等の難しさを知ることができた。

【 桐山 大司 】

自分で制作したものをプログラムで制御するという貴重な経験ができた。プログラムはとても難しかったが、望む動作をした時には大きな達成感が得られた。

非接触 IC カードと Raspberry Pi を用いたシステムの構築

研究者：後藤 太聖・宮川 託実

1 はじめに

Raspberry Pi と非接触 IC カードについて知り理解を深めると同時に、システムを構築することを目的として研究を行った。

本研究ではサーボモータで鍵を制御するシステムと入退出管理システムの2つを構築した。また、非接触 IC カードは FeliCa を用いた。FeliCa とは、Sony が開発した非接触 IC カードの技術方式及び登録商標の名前である。以下、非接触カードを FeliCa とする。

2 研究内容

本研究では、FeliCa の ID を読み取る機器である PaSoRi とシングルボードコンピュータである Raspberry Pi を用いて、FeliCa の固有 ID 読み取り、サーボモータの制御、入退室のデータベース化に取り組んだ。

(1) FeliCa の読み取り

Raspberry Pi と PaSoRi を USB で接続し、nfcpy PythonModule をインストールして、FeliCa をかざすことで固有 ID を読み取るようにした。



図1 FeliCa を PaSoRi にかざす様子

FeliCa には製造時 IC チップに記録されている書き換え不可能な固有 ID 暗号が存在する。



図2 ID 読み取り成功画面

図2の下線のように 16 桁の英数字で表されているものが ID である。

(2) サーボモータによる鍵の制御

読み取り成功した ID を、事前に登録されている ID であるかを判断し、一致すれば FeliCa に登録した名前の「Ogaki Taro」がターミナルに表示され、サーボモータが動き、施錠、開錠されるプログラムを Python で作成した。また、読み取った

ID が登録したものと違う場合、「Not Authorized」と表示されるようにした。(図3)

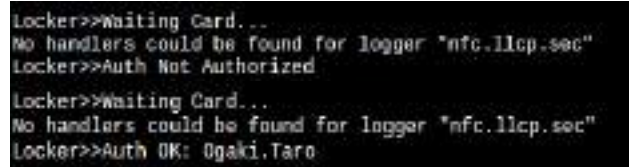


図3 FeliCa を読み取った結果
(上：未登録 下：登録済)

次に、Raspberry Pi でサーボモータを制御させるために Raspberry Pi についている GPIO ピンを用いた。



図4 Raspberry Pi の GPIO に接続した様子
(3) SQLite について

SQLite は、小型で処理が速く、使用者一人のみで問題解決ができ、機能性の高いリレーショナルデータベースである。

今回は、Python で SQLite のデータベースを作成した。テーブル内のカラムに、レコードとして、生徒の氏名、ID、時間、生徒の状況(「入室」または「退室」)を格納したデータベースを作成した。

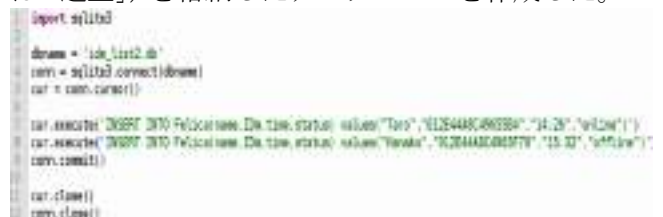


図5 SQLite のデータベース作成プログラム
(4) 入退室管理システムの構築

FeliCa を PaSoRi にかざすと、入退室した生徒の氏名、入退室した時間と状況を管理できるプログラムを Python で作成した。このプログラムでは、テーブル内において、事前に登録されている生徒の状況が、「入室」である場合、FeliCa をかざすと「退室」に変更され、「退室」である場合、「入室」に変更される。また、それぞれ入室した時間、退室した時間が記録される。


```

class FelicaIO:
    def __init__(self, gpio):
        self.gpio = gpio
        self.pwm = PWM(gpio.PIN_18, 100)
        self.pwm.start()
        self.pwm.write_duty_cycle(100)
        self.gpio.set_mode(gpio.PIN_18, gpio.OUT)
        self.gpio.set_direction(gpio.PIN_18, gpio.OUT)
        self.gpio.set_pull_up_down(gpio.PIN_18, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_19, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_20, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_21, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_22, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_23, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_24, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_25, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_26, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_27, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_28, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_29, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_30, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_31, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_32, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_33, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_34, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_35, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_36, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_37, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_38, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_39, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_40, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_41, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_42, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_43, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_44, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_45, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_46, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_47, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_48, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_49, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_50, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_51, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_52, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_53, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_54, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_55, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_56, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_57, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_58, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_59, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_60, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_61, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_62, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_63, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_64, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_65, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_66, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_67, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_68, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_69, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_70, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_71, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_72, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_73, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_74, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_75, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_76, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_77, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_78, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_79, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_80, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_81, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_82, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_83, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_84, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_85, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_86, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_87, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_88, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_89, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_90, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_91, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_92, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_93, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_94, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_95, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_96, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_97, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_98, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_99, gpio.PULL_UP)
        self.gpio.set_pull_up_down(gpio.PIN_100, gpio.PULL_UP)

```

図6 入退室管理システムプログラム

```

pi@raspberrypi:~$ sudo python3 felica_io0115.py
カードをタッチしてください
読み取り中
カードのID: 012e44a8c49655b4
Taroさん
22:13
退室しました。
カードをタッチしてください
読み取り中
カードのID: 012e44a8c49655b4
Taroさん
22:13
入室しました。
カードをタッチしてください

```

図7 実行画面

3 研究過程

- 4～6月 : 調べ学習
Raspberry Pi の初期設定
PaSoRi の初期設定
- 7、8月 : システム作成開始
FeliCa の読み取り
SQLite のデータベース作成
サーボモータのプログラム完成
- 9、10月 : FeliCa 判別プログラム作成
入退室管理システムの構築
展示用ドアの製作
- 11月 : 文化祭展示
文化祭ポスター作成
- 12月 : レポート作成開始
- 1月 : 資料の作成と発表

4 研究成果

(1) PaSoRi で FeliCa の読み取り

Raspberry Pi での初期設定や Python、nfcpy などのインストールの方法をインターネットで調べ参考にすることで、FeliCa の読み取りに成功した。自力でこのプログラムを作成するのは大変難しいことだということが分かった。

(2) Raspberry Pi でサーボモータの制御

Raspberry Pi の GPIO ピンには、それぞれに役割があり、理解することができた。また、モータ

ーに関する知識も学ぶことができた。今回の制御では、パルス幅の高さを変える PWM と、パルス信号のオンオフの比率を変更するデューティ比の2つを理解し制御できた。

(3) SQLite のデータベースの制御

テーブルに格納されているデータを更新するために、Python のプログラム内で SQL の update 文を用いた。結果、テーブルが更新され、入退室管理をすることができた。

5 課題

サーボモータによる鍵の制御での課題は、まだ実用的なレベルではないというところである。スマートフォンなどの他種デバイスとの連動を実現するべきだ。

また、SQLite のデータベース作成の課題は、現在のデータベースの表示画面は少し見にくいというところである。そこを修正して見やすくするべきだ。

6 チームの感想と反省

【後藤 太聖】

4～6月の間は Raspberry Pi と FeliCa について知識を深め、自分のやってみたい研究内容を探した。その時に、Raspberry Pi にはできることが数多くあるということを知り、その時はどんなものを作ろうかと少し楽しみながら考えていたが、Raspberry Pi 内でシステムを作るには Python が欠かせないということを知り、言語の習得に対して少々不安が生まれた。ところが、構文が意外とシンプルで分かりやすく、プログラムを完成させることができた。本研究でできた経験は良いものだった。



【宮川 託実】

Python や SQLite を初めて使用したため、いくつものエラーメッセージの解説に苦戦したが、調べ学習を行ったことや担当の先生に、わからないことを教えていただき、多くの問題点を解決しながら研究を進めることができた。また、SQLite のデータベースの作成は非常に簡易ではあったが、Python との接続が上手くできず少々時間を費やしすぎてしまったため、さらに、Python と SQLite について学ばなければならないと実感した。本研究では、FeliCa と Raspberry Pi の仕組みなどを学び、理解を深めることができ、研究で得たことを今後の人生でも生かしていきたいと思った。



情報技術科 PR 活動

研究者：北嶋 遙、森 星也、森 優貴

1 はじめに

中学生に対して来年度の情報技術工学科を PR するポスターや動画、アニメーションを制作した。新しく生まれ変わる情報技術工学科について興味をもってもらい、より多くの新生が入ってくるような PR をすることが目的である。

2 研究内容

(1) ポスター作成

夏休みの高校説明会や 10 月の中学生 1 日入学などで掲示する、学科の PR をするポスターを作成した。また、自らの実力を伸ばすためにポスターコンクールの応募作品も作成した。

作成には ibisPaint というアプリケーションを使用した。フィルター加工や、エフェクトを使うことができ、通常のイラストが明るく、鮮やかに作成することができた。

(2) 動画制作

夏休みの高校説明会に参加した中学生に情報技術工学科について興味を持ってもらえるように情報技術科の 3 年生を対象としたインタビュー動画を制作した。高校生生の声を聞くことができ、実際の映像を見たり実体験を聞いたりしながら学科の概要を知ることができるため、より印象に残りやすいと考えた。

(3) アニメーション制作

10 月の中学生 1 日入学に参加する中学生に対して、情報技術工学科の PR を行うアニメーション「THE LIFE」を制作した。物語の内容は、情報技術工学科に入学した情報系大学への進学を目指す女子生徒と、IT 企業への就職を目指す男子生徒がそれぞれの苦難を乗り越え夢を叶えることができたというものである。Official 髭男 dism 様の「宿命」という楽曲を使用させて頂き、それに合わせたアニメーションを制作した。制作はシナリオ考案、絵コンテ、線画、配色、編集、修正の順で行った。

3 研究過程

- 4 月～ 計画書の作成、
コンクールへ向けた作品の作成
高校説明会で公開する動画の構成
- 6 月～ アニメーションの構想・楽曲決め
動画の方針決定
ポスター作成・キャラクターデザイン
絵コンテの作成
- 7 月～ 動画の撮影・編集

アニメーションの制作

- 10 月～ アニメーションの細部確認
- 11 月～ 文化祭に向けたポスターの作成
- 12 月～ レポートの作成
- 1 月～ 資料の作成と発表

4 研究成果

(1) ポスター

コンクールや、情報技術科の PR に向けたポスターを作成した。ポスターはコンポジション（構図や構成）、文字の大きさや位置などで見え方や伝わり方が変わるのでと理解した。



図 1 学科群・学科ポスター

ポスターコンクールの結果として、日本国際ポスター美術館主催 第 18 回全国高校生ポスターコンクール/テーマ「祭」では入選、愛知県印刷工業組合主催 第 10 回ポスターグランプリ/テーマ「輪」では特別賞・岐阜県教育委員会賞を受賞することができた。



図 2 入選・特別賞を受賞した作品

(2) インタビュー動画

当初、プレゼン動画を制作した。しかし、最初に制作した動画は情報量が多く、効果的な動画とは言えなかった。そこでインタビュー動画に路線変更をすることにした。

編集では、Adobe Premiere Pro CC 2018 を使用し、高校説明会を訪れる中学生に対して公開する動画を制作した。内容は情報技術部の活動、課題研究、デザインの学習やセンター入試の対応もできるということである。親近感を覚えてもらうために学年と氏名のほかに、出身中学校のテロップを画面右側に挿入した。



図3 インタビュー動画

(3) アニメーション「THE LIFE」

全てを1から考案し作画枚数468枚、2分19秒の完全オリジナルアニメーションが完成した。

シナリオを考案する際、情報技術工学科の魅力や日常を伝えつつ、自分の力で夢を叶える物語にしたいと考えた。学科PRではあるが、夢は自分の力で叶えるということを強調したかった。また、情報技術工学科への入学は、夢を叶えるための路の途中であり、「THE LIFE」には自分の人生をどう歩むか考えて欲しいという想いを込めた。

編集には、Adobe After Effects CC 2018 を使用し、作画を終えたものから順に編集を行った。他にも光や降雪などのエフェクトの追加、3D機能を利用した編集も行った。



図4 アニメーションの編集画面

無事に完成し中学生1日入学の時に公開することができた。その後、先生方やクラスメイトからの意見を参考に表現や人物の動きなどの修正、シーンの追加などを行い、文化祭展示用のアニメーションを制作した。



図5 アニメーション「THE LIFE」

5 考察

今回の研究ではポスターの作成や動画の編集など、研究者それぞれの得意な分野を生かしたPR活動ができた。一方で、それぞれが独立したことをしていたために、知識に偏りができてしまったことが反省点である。

6 まとめ

(1) 成果

情報技術科のPRというテーマのもと、動画やアニメーションの制作や、ポスターの作成など様々な方法でPR活動を行うことができた。

アニメーションを上映した際は、興味を持ち、見入っている中学生の姿もあり、その後のアンケートにアニメーションについて学びたいと回答してくれた中学生もいた。

(2) 課題

ポスターを作る際に、誰に向けたポスターなのかを明確にしていなかったことで何度も指摘を受けた。インタビュー動画では、見やすい構図がまだわかっていない。これらのことはとても重要だが、今回の研究では疎かにしてしまった。誰に向けてどんな作品を作るのかを最初に設定すること、誰にでも見やすい表現をすることを意識して作品を作ることが課題である。

7 感想

[北嶋 遥]

アニメーション制作は初めてなので、しっかり完成させられるか不安だったが、チームで協力して形に出来たので良かった。自分の役割を果たし、楽しみながら作成することができ、達成感があった。

[森 星也]

今回、アニメーションやポスターを作ってみて、一人では何もできないと実感した。いろいろな作品を完成してみて、この課題研究を通して学んだ技術と知識を自分の将来にいかしていきたいと思った。

[森 優貴]

動画やアニメーションの制作の知識がない状態からのスタートでとても苦労したが、貴重な体験だったため楽しくでき、達成感があった。

FPGA による迷路ゲーム器の製作

研究者：伊藤 伸
伊東 巧成

1 はじめに

今回の研究における達成目標は、FPGA を使って 2D と 3D の両方での迷路ゲーム制御回路を製作することだ。迷路ゲームを製作するにあたっては、他にないオリジナリティ溢れた要素を取り入れて製作することで、FPGA への理解を深め、技術習得をすることを目的として研究を進めた。

2 研究内容

本研究では、FPGA と VHDL 言語を利用して、モニタ画面に出力される迷路ゲーム器の製作をした。リセットするごとに乱数によって新しい迷路パターンが生成され、さらにキャラクターの目線に立ってゲームが出来る機能を実装した。図 1 は今回使用した FPGA ボードで、どちらも Xilinx Spartan-3E やスイッチ、LED、画像出力用の VGA 端子があり、研究を進める中で容量が多い右のものに移行した。



図 1 2つの FPGA ボード

FPGA とは、ユーザーがハードウェア記述言語を使って動作をプログラムできる論理回路の集まり（集積回路）である。

VHDL とは、そのハードウェア記述言語の 1 つであり、それを用いてスイッチの入力、迷路の画像の描画、キャラクターの移動等をプログラムした。

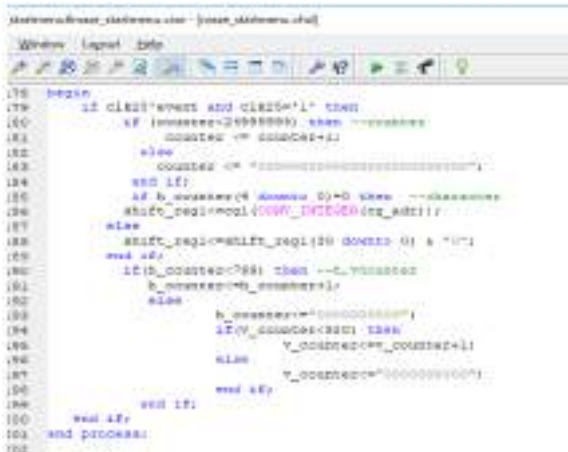


図 2 迷路設計プログラムの一部

3 研究過程

- 4月 VHDL 基礎学習
一年の到達目標設定
- 5月 部品の発注
画面表示の技術習得
- 6月 基本迷路画面の出力
- 7月 キャラクターの作成
ボタンに対応したプログラムの作成
試作版の迷路パターン設計
迷路の自動攻略プログラムの作成
- 8月 新しい迷路設計プログラムの作成改良
迷路の 3D 表示プログラムの作成
ストップウォッチプログラムの作成
- 9月 迷路を文化祭発表用に調整
迷路の自動生成プログラムの作成
キャラクターの移動方法の改良
リセットボタンの実装
ゴール判定実装
スタートメニュー実装
- 10月 ゴール後の再スタート機能実装
操作用コントローラ作成
- 11月 文化祭展示用ポスター作成
- 12月 レポート作成
- 1月 資料の作成と発表

4 研究成果

(1) 迷路の自動生成

製作当初は迷路のパターンをデータとして与えていたが、スタートメニューを表示した時には迷路が自動的に生成されるようにした。迷路をクリアした後や、途中で迷路脱出を断念しリセットボタンを押した場合でも、違う迷路が生成される。

自動生成の方法は次のとおりで、まず、すべてのますの四方を壁に初期化し、壁を掘り進めるように道を生成していく。掘り進める方向は乱数を用いて決定されるため、二度として同じ迷路が生成されることはない。また、迷路の自動生成においてループや使われない領域が出ないように条件を設定し、スタートからゴールへの最短経路が 1 通りだけ存在する迷路を生成するように工夫した。

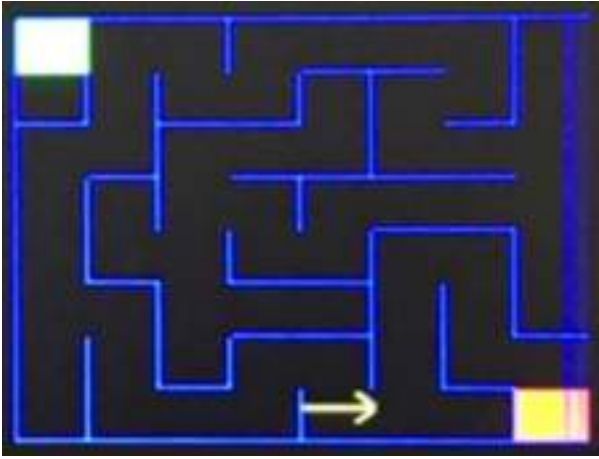


図3 自動生成された2Dの迷路画面例

(2) 迷路の自動攻略

設定されたボタンをONにすることで、自動で迷路を攻略するようにした。自動生成された迷路をコピーし、行き止まりを塗りつぶすことで、スタートからゴールまでの最短経路を生成している。その最短経路を元にキャラクタを動かし、ほんの数秒でクリアする。

(3) 2Dと3D両方の同時画面表示

画面上に2Dと3Dで両方の迷路が表示されるようにした。3Dでは、キャラクタの目線に立っているかのような一人称視点で表示される。キャラクタがいる周りのマス情報を元に、立体的に見えるように線を描画している。右上には、迷路とキャラクタを上から見ており、3D・2Dどちらのキャラクタも同期して動くようになっている。2Dでの迷路表示は、キャラクタが通った道のみを表示することもできるようにし、ゲーム性を高めた。

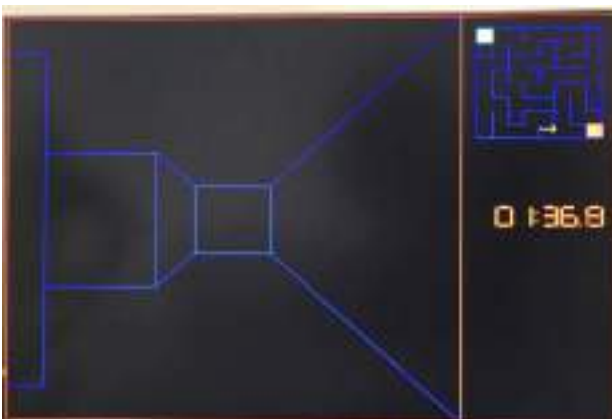


図4 画面表示された迷路例

(4) コントローラでの操作

市販のモニタ台を改造して押しボタンを取り付けて、FPGAで表示された迷路のスタート、リセット、そしてキャラクタの操作をできるように

した。ボタンに使用するリミットスイッチにリード線をはんだ付けし、ボードに付いている拡張端子のピンヘッダーと接続することで、FPGAにボタンが押されたことを知らせることができる。

また、電子ブザーを利用した効果音発生回路を製作し、これもFPGAボードの拡張端子に接続し、ボタンを押したときの反応音が出るようにした。

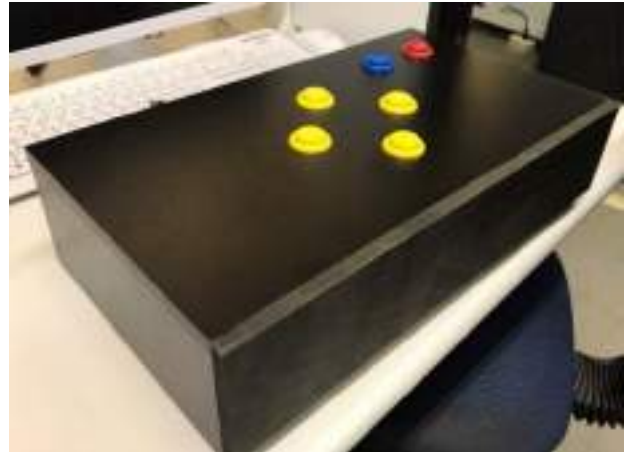


図5 コントローラ

5 考察

今回の製作では、当初ボタンを押してキャラクタを迷路上で動かすとき、少しの押し込みでも2回同じ行動を繰り返してしまうチャタリングという現象が起きていた。その対処として、自己保持回路を用いて、ボタンの1度の押し込みに対して1度しか反応しないようにすることができた。

しかし、まだ改良したい点もある。迷路の途中にカギなどのアイテムを配置し、カギを拾うことでゴールできるようにするなど、さらに発展的な機能を実装することでゲーム性を高めたい。

6 まとめ

(1) 成果

今回の研究では迷路ゲーム器の製作を通じてFPGAやVHDLへの理解を深め、技術を習得することができた。

3Dでの迷路表示と2Dでのミニマップ表示を行うことで、臨場感のあるゲームにすることができた。また、迷路の自動生成と自動攻略の実装に成功した。コントローラを製作することで、加工技術高めるとともにゲームにおける操作性を高めることができた。

(2) 課題

チャタリングを解消したことでボタンの長押しによる連続移動機能が実装できなくなってしまった。ゲーム性を高めることを目標としていたので機会があれば改善したい。

FPGAによるピンポンシミュレータの製作

研究者：荒川 楓真 加納 直樹
坪井 星風

1 はじめに

本研究における達成目標は2つあった。

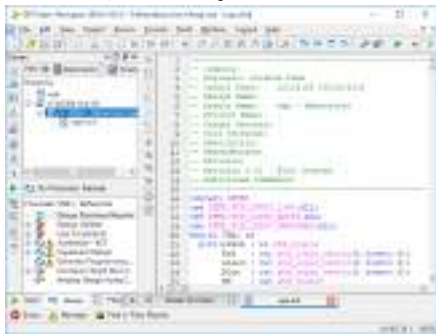
1つは文化祭までにピンポンシミュレータとして、基本的な動作をするものを製作し、来場者に楽しんでもらうことである。

もう1つは文化祭後にプログラムを発展させ、様々なゲーム的要素を追加したオリジナルのピンポンシミュレータを製作することである。

2 研究の内容

VHDL 言語を使用し、FPGA を用いたピンポンシミュレータと操作のコントローラを製作した。

VHDL はハードウェア記述言語(HDL)の一種で、主として論理回路、特にFPGAやASICなどの設計で使われるものである。プログラムの開発にはXilinxのISEを使用した。



開発環境 Xilinx ISE

FPGA は Field Programmable Gate Array の略で即座にその場でHDLを使ってプログラム可能な論理回路デバイスで、このFPGAが搭載されたBASYS2という基板を使用し研究を進めていった。



BASYS2 外観

この基板にはFPGA Spartan-3Eのほか動作テスト用のスイッチやLED、ディスプレイ接続用のVGA端子などが載っている。

3 研究過程

- 4～6月：VHDLによる画像表示演習
- 7～8月：ブロック崩しの製作
- 9～11月：ピンポンゲームの製作
 - ブロック崩しの発展
 - コントローラの製作
- 12～1月：プログラム修正、追加

4 研究の成果

(1) VHDLによる画像表示演習

最初にVHDLの使い方と画像表示回路の仕組みについて理解するために演習を行った。国旗表示や秒カウンタの製作、押しボタンスイッチのオンオフによる表示文字の切り替え等を行った。

これらの演習から、VHDLの基本文法と液晶ディスプレイに画像を表示させる方法について理解を深めていくことができた。

```
process (clk25)
begin
  if clk25'event and clk25='1' then
    if (h_counter<799) then
      h_counter<=h_counter+1;
    else
      h_counter<="0000000000";
      if (v_counter<520) then
        v_counter<=v_counter+1;
      else
        v_counter<="0000000000";
      end if;
    end if;
  end if;
end if;
end process;

HS<="0" when h_counter>=640+16 and h_counter<640+16+95 else '1';
VS<="0" when v_counter>=480+10 and v_counter<480+10+1 else '1';
blank<="0" when h_counter>=640 or v_counter>=480 else '1';

rgb<="000" when v_counter<160 else "111" when v_counter<320
else "001";
```

画像表示制御のVHDLプログラム



オランダ国旗の表示



文字Hの表示

(2) ブロック崩しの製作

ピンポンシミュレータの元となるブロック崩しの製作をした。パドルを左右に動かしてピンポン玉を跳ね返し表示されているブロックを消していくものである。これにより一通りブロック崩しの各動作の仕組みとプログラムの構造について理解できた。



プレイ画面

(3) ピンポンゲームについて

ブロック崩しで得た知識を使い1対1で遊ぶことができるピンポンゲームを製作した。



ピンポンゲーム
スタート画面

ピンポンゲーム
プレイ画面

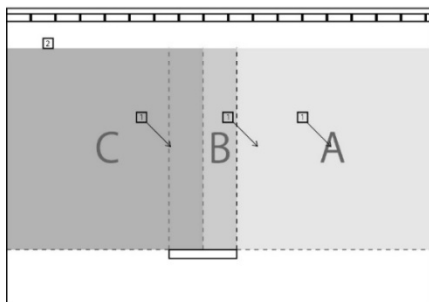
パドルの形を縦長に変更し、当たり判定や移動の制限を設けた。また、ピンポン玉が各画面端に当たると表示が消え、点数がカウントされる仕組みとなっている。ゲームの残り時間を示すタイマーと得点のカウントとの組み合わせから勝敗が決まるようにした。

(4) ブロック崩しの発展

ブロック崩しの発展として、様々な機能を追加した。2つのピンポン玉に対するパドルの自動操作機能では、パドルとピンポン玉の位置関係を12通りのパターンに分け、それぞれに決められた動きを与えた。

たとえば、ピンポン玉1の座標がピンポン玉2より低かつパドルより高い場合には

- ・パターンA：パドルの座標に6足し続ける
 - ・パターンB：パドルの座標に1足し続ける
 - ・パターンC：パドルの座標に6引き続ける
- といった動作をプログラムした。



```
if v_dir='1' and v_pos>v_pos2 then
  if h_dir='1' then
    if pad_h_pos+pad_h_size<=h_pos+10 and
      pad_h_pos+pad_h_size<640 then
      pad_h_pos<=pad_h_pos+6;
    elsif pad_h_pos+pad_h_size>h_pos+10 and
      pad_h_pos+pad_c_size<h_pos and
      pad_h_pos+pad_h_size<640 then
      pad_h_pos<=pad_h_pos+1;
    elsif pad_h_pos+pad_c_size>h_pos+10 and pad_h_pos>6
      then pad_h_pos<=pad_h_pos-6;
    end if;
  end if;
end if;
end if;
```

VHDLの自動操作制御部分

(5) コントローラの製作、機能

基板上の押しボタンスイッチは小さいので文化祭での実演に向けて専用のコントローラを製作した。MDF材のケースに穴あけを行い、そこにジョイスティックや押しボタンスイッチを取り付け、コントローラの裏側で配線を行いBASYS 2に接続した。

パドルはジョイスティックを上を倒すと前進、下を倒すと後退、右を倒すと右に前進、左を倒すと左に前進する。4つの押しボタンはゲームの開始などそれぞれ別の機能を持たせた。



コントローラ



裏側の配線

5 まとめ

(1) 成果

普段は触れることのないVHDLやFPGAに触れることでハードウェア・ソフトウェアの分野らについて多くのことを学び、また深めることができた。

(2) 課題

ピンポンゲームでは、文化祭展示の段階では動作エラーが多く修正するのに時間がかかった。追加機能としてタイマーや点数のカウントのプログラムを製作したが、点数が入るごとにタイマーをリセットすることがどうしてもできなかった。パドルの自動操作中にパドルとピンポン玉の跳ね返る角度が逆になってしまうバグやパドルの移動中はピンポン玉が跳ね返らないなどの不具合が残っている。

また、当初はコントローラにプログラム自体の「ON/OFF」機能を持たせたスイッチを用意していたが、ケースのパネルが厚くて取り付けるができなかった。

6 チームの感想

【 荒川 楓真 】

FPGA、VHDLを初めて使い、全く知らないことで最初は戸惑ったが、演習や研究を行っていくうちに、だんだんわかるようになった。

【 加納 直樹 】

初めは、課題研究の期間が1年間と聞いて長いと思ったが、実際に課題研究が始まるとやることが多く、この1年間はとても短く感じた。

【 坪井 星風 】

VHDL言語を初めて使用してプログラミングを行った。最初はわからなかったが扱っていくうちに慣れていくことができた。

LED イルミネーションの製作

研究者 馬淵将大 渡部俊之輔 アルベスジョナタス

1 はじめに

今回の研究での達成目標は、3年間で学んだことを活かし、LED イルミネーションを製作し、大垣市の展示会に展示することで大垣市への地域貢献をすることです。

2 研究の内容

本研究ではまず、LED を順番に点滅させるための Arduino でプログラム作成、pcbe を使った基板のパターン設計および基板の製作、そして展示する舞台が大垣ということで松尾芭蕉と水門川をモチーフにした板を製作しました。



完成した LED イルミネーションパネル

3 研究過程

4・5月…イルミネーションのデザイン案を考える。

6月………イルミネーションの点滅方法を考える

7月………文化祭用のイルミネーションの展示案、部品を考える。

8・9・10月

文化祭用のプログラム・基板の製作

11・12月…大垣イルミネーション展示用の製作・仕上げ

4 研究の成果

(1) pcbe について

pcbe では基板のパターン設計を行いました。基板作成では2種類の基板を設計しました。1つは LED や抵抗などおおもとの基板を設計し、もう1つは基板を収束する基板です。

収束基板を作った理由としては Arduino の使えるピンが6ピンのみしかなかったためまた配線をまとめるためにも収束用の基板を製作しました。

文化祭用の基板を作成したときに銅線が多くなってしまったため、大垣のイルミネーションを作る際には銅線をできるだけ減らすためにも銅箔をつなげるようにしました。



pcbe パターン設計

(2) 基板の製作について

pcbe で作成したパターンを参考に基板を製作しました。エッチングでは、ポジ感光基板に現像しました。自分たちが作る基板の量がかかり多かったので、1枚1枚エッチングし、現像するのが大変でした。

エッチングが完成したら、穴あけ、はんだ付けを行いました。三人で分担して効率よく行いました。

基板同士を接続する際に、エッチングで現像されなかった部分などを見つけ LED が正常に光るよう修正していきました。



(3) プログラムについて

```
tameshi2.ino | Arduino 1.8.8
ファイル 編集 スケッチ ツール ヘルプ

tameshi2.ino
void setup() {
}
void loop(){
  int i=3;
  pinMode(i,OUTPUT);
  for(int j=0; j<=255; j++){
    analogWrite(i,j);
    delay(5);
  }
  for(int j=255; j>-1; j--){
    analogWrite(i,j);
    delay(5);
  }
  analogWrite(i, 0);
  delay(225);

  i=5;
  for(int j=0; j<=255; j++){
```

このプログラムは私たちが今回自分たちで作成し、使用したものです。

最終的には製作した基板とつなげて LED を順番に点滅させました。

プログラムは Arduino IDE というソフトウェ

アを使ってプログラム作成をしました。最初は digitalwrite 文を使ってプログラムを行っていましたが、ただついて消えるだけの digitalwrite 文より、analogwrite 文のほうが明るさ調節することができ、よりイルミネーションらしい点滅ができるということで analogwrite 文を使いました。

5 まとめ

(1) 成果

これまでの実習で得た基板のつくり方などを活かし自分たちで一からイルミネーションを製作することができた。また、Arduino という使ったことのないものも上手く利用し、作品にあった LED 制御ができた。

(2) 課題

今回の課題研究での課題点は、川に使った LED の基板をただビニールで覆っただけの防水加工しかできてなく、防水にしても見映え的にも悪かったことです。樹脂ポッティングなどももう少し考えて工夫したかったです。

6 チームの感想

【アルベス ジョナタス】

Arduino は初めてだったけど慣れない中で自分達の思うプログラムを一から作成できてよかったです。

【馬淵 将大】

今までの実習で学んできた pcbe やエッチングを思い出しながら基板を一から作成しより理解を深められたと思いました。

【渡部 俊之輔】

スロースタートだったため、もう少し計画通りに進めればもっといい作品ができたと思いました。

3Dプリンタの活用研究

研究者：問山 晴生

1 はじめに

私は今まで実習などを通してものづくりの難しさを痛感した。そのため、設計製図の作成に重きを置き、3Dプリンタを用いて、描いた図面(立体物)を成形したいと思い、この研究を選択した。

2 研究の内容

3Dモデリングソフト Autodesk fusion360 について基本操作を学びながら、立体物を描き、3Dプリンタで、3DCADなどの3次元ソフトウェアで設計された3次元データを元に断面形状を積層し、立体物を成形する。

3 研究過程

- 4、5月 : 計画の立案、調査研究
- 5、6月 : ペン立て、コマの作成
- 7、8月 : スマホスタンドの作成
- 9～11月 : 東京オリンピックのロゴ作成
- 11、12月 : 3Dプリンタで造形物の成形
- 1月 : レポートの作成と発表

4 研究の成果

(1) ペン立て、コマの作成

私は fusion360 の使用方法を理解していなかったため参考図書で図形的设计を行った。

fusion360 でペン立てを設計するにあたり、はじめに、平面のスケッチを作成した。その後、押し出しコマンドを使って平面の図面を立体にした。立体の図形を作っても、ペンはいれられないため、設定で切り取りを使用すると中をくり抜くことができた。そして見栄えを整えるとペン立てができた。

コマの作成では、学んだコマンドは回転コマンドである。まず断面の図面を作成し、その後回転コマンドを使うと先ほど設計した断面が360度回転した図形を設計できる。



図1 ペン立て

図2 コマ

(3) スマホスタンドの作成

スマホスタンドでは寸法拘束、形状の押し出しや修正コマンドの使用方法について学んだ。ここまでは、書籍を参考に作業し、fusion360 についての理解を深めることができた。

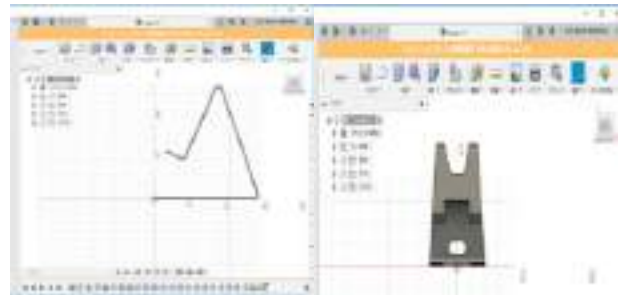


図4 側面図

図5 正面図

(2) 東京オリンピックのロゴ作成

今年開催される東京オリンピックのロゴの作成を行った。これまでは参考図書を使って設計を行ってきたが、ここではロゴ作成サイトを参考にして図面を設計した。作成手順は下記のとおりである。

- ① 正12角形を作成する
- ② 各辺の中点をとる
- ③ 半分の長さからなるひし形を作成する
- ④ 正方形に近い図形を作成する
- ⑤ 残ったところに正方形を書く
- ⑥ ひし形中点をとる
- ⑦ 中点つないで細長い長方形を作っていく
- ⑧ 中点をつないで長方形を作っていく
- ⑨ とった中点をもとに正方形を作っていく
- ⑩ 余分な線を消していく

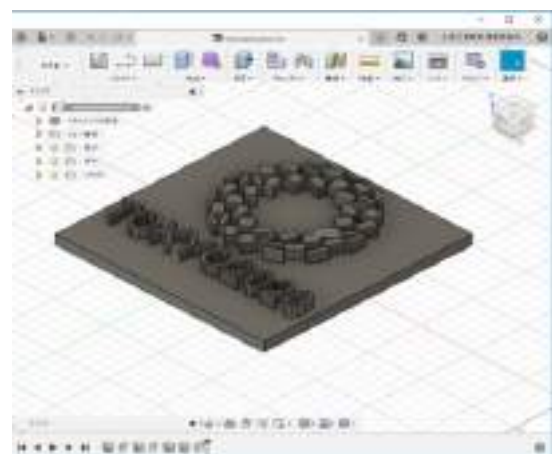


図6 東京オリンピックのロゴ

(4) 3Dプリンタについて

3Dプリンタは設計した fusion360 でのデータをもとに、人がいなくても自動で立体物を成形する。自らの手や工具などを使用して作品を製作することより、ミリ単位で正確にものづくりができるため非常に便利である。成形する工程を図7から図10に示す。

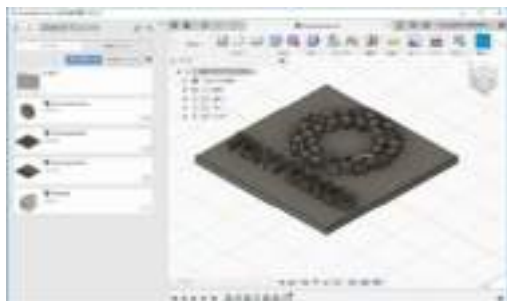


図7 3Dモデリングソフトで造形物の設計



図8 スライサーを用いてGコードに変換

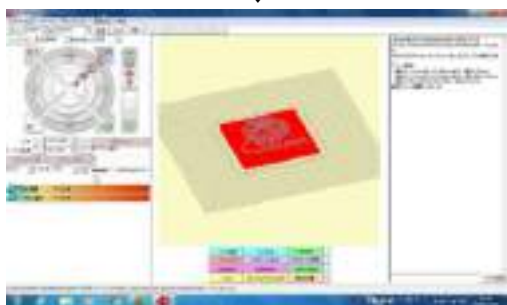


図9 制御ソフトを用いて3Dプリンタに出力

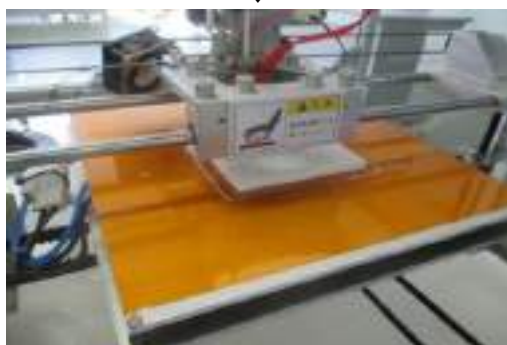


図10 出力中の東京オリンピックのロゴ

(5) 作品



図11 文化祭に展示した作品

fusion360 を使いその後3Dプリンタを用いて造形物を成形した。コンピュータなどのスペックなどの仕様上、成形する範囲は限られていたが、実際に自分の思い描いた作品を作ることができた。

5 まとめ

(1) 成果

最初は使い方など全く分からずに本当に作品を作ることができるのか不安だったが、fusion360 の基本操作について理解することができ知識を増やすことができた。

また、立方体の中心をもつて動かすと様々な方向から図形を見ることができることに気づき、より正確に図面を設計することができた。文化祭の時に作品を展示できた。

(2) 課題

・コマンドに対して、まだ理解が浅いことである。設計をするにあたり、様々なコマンドがあるが複雑な図面や曲線などに対して、まだ自分で選んでコマンドを使うのが難しいため、コマンドの内容を覚えて取り組みたい。

・寸法に関しても複雑な図形や小さい図形の角度を測るときなどは、非常に時間がかかってしまうことである。また、パソコンがフリーズすることがあった。必要のないところの寸法を測らないようにするか、別のコマンドで長さを合わせるなど工夫して取り組みたい。

6 感想

課題研究を通して fusion360 の使い方や基本的な知識を学ぶことができた。また、複雑な作品を作成することができたことが、大きな自信に繋がった。

現在、科学技術や医療などの分野で使われている3Dプリンタを使用して、自分で考えた立体物が形になって嬉しく感じた。この課題研究で得た体験を今後に活かしていきたい。

Raspberry Pi によるフィジカル・コンピューティング (サーバ構築、顔認識)

研究者：近藤 拓海

：中島 暢慎

1 はじめに

本研究の達成目標は、2つある。

1つ目は、顔認識をした座標をもとに、Raspberry Pi に接続したモータを制御して、追尾式のカメラを目指す。理由としては、現在人口知能は急速に発展しており、その中でより精度の高い処理を行うためにはどうすればよいか知りたかったためである。

2つ目は、サーバの構築やリモート通信を通して Raspberry Pi の基本的な知識・技術の習得をする。理由は、Raspberry Pi は教育用途だけでなく、産業用途でも使われるようになったことで今後 Raspberry Pi の基本的なことを知っておくべきだと考えたためである。

2 研究の内容 (中島)

私の研究ではリモート通信の1つに VNC 接続を使った。VNC 接続とは別のコンピュータの画面を自分が使っているコンピュータに表示させるというものである。

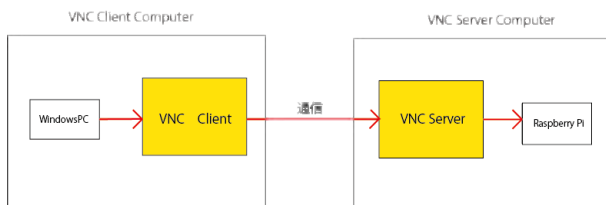


図1 VNC 接続の例

サーバ構築では DLNA というサーバを研究した。DLNA とは家電やタブレット端末等 LAN を通じて動画・音楽・写真をやりとりできるようにするためのガイドラインのこと。Raspberry Pi に入っている音楽や写真を Windows で表示させた。

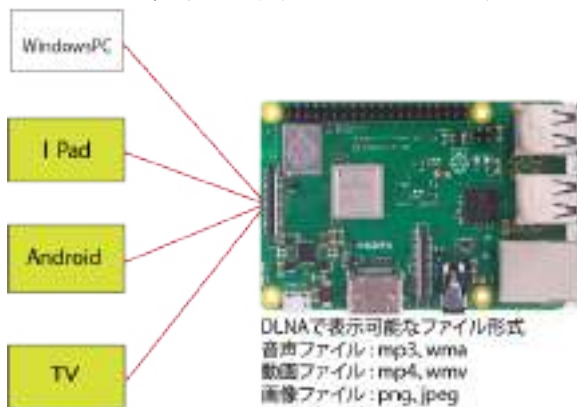


図2 DLNA サーバの構築例

3 研究過程 (中島)

- 4月 テーマ設定
計画書の作成
- 5月 研究内容に関する調査・研究
- 6月 Raspberry Pi の初期設定
- 7月 SSH・VNC 接続
- 8月 samba の設定
- 9月 apache に関する調査・研究
- 10月 DLNA に関する調査・研究
- 11月 apache サーバの構築
DLNA サーバの構築
- 12月 レポートの作成
- 1月 資料の作成と発表

4 研究の成果 (中島)

本研究で Raspberry Pi を使うことで早い段階で Linux についての使い方や知識を学ぶことができた。様々な設定をしていくことで Linux コマンドやファイルマネージャの仕組みが分かった。また、VNC や SSH 接続をしたことで一部のポート番号の働きを理解することができた。

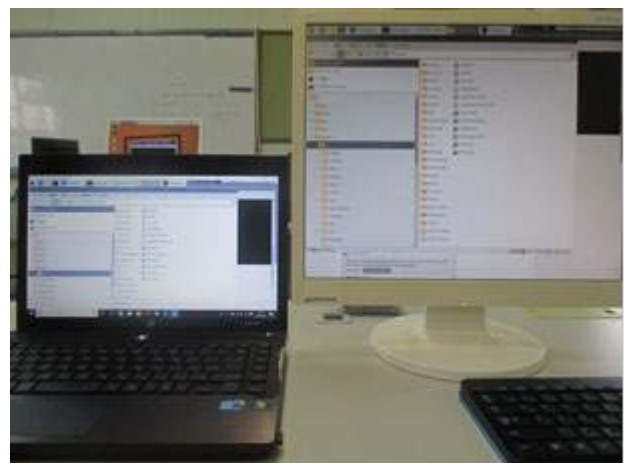


図3 VNC 接続した画像

また、DLNA を使ってスマホに入っている音楽を、ワイヤレスでオーディオ機器から再生することや、録画した番組を他の部屋にあるテレビやタブレット端末で視聴できるようになり、音楽・ビデオプレーヤにすることができた。

Raspberry Pi を研究したことで日常生活に役立てること増え、自分の目標を達成することができた。

5 研究内容（近藤）

私の研究では OpenCV というライブラリを使用して、画像処理を行い、顔認識をして顔の座標を出力するプログラムを作成する。



図4 開発環境

本研究で使用したプログラム言語は「Python」である。なぜならば、様々なライブラリが充実しているからだ。また、近年とても注目されている言語でもあるため、習得したいという気持ちは強くあった。

```
#coding: utf-8
import cv2
import sys
cc = cv2.VideoCapture(0)
rr, img = cc.read()
cv2.imwrite('pi.jpg',img)
    中略
cascade_file = "haarcascade_frontalface_alt.xml"
image_file = "pi.jpg"
cv2.imwrite("kao.jpg", img)
```

図5 制作したプログラム（中略）

```
pi@pi:~$ python3 kao.py
Corrupt JPEG data: 1 extraneous bytes before marker
xd0
Face recognise 顔認識
pi@pi:~$ python3 kao.py
Corrupt JPEG data: 2 extraneous bytes before marker
xd0
顔の座標 = 184 50 338 338
```

図6 プログラム実行結果

6 研究過程（近藤）

- 4月 計画書の作成（何を作るか）
- 5月 計画書の作成（どう作るか）
Pythonの学習開始
- 6月 深層学習に関する調査・研究
- 7月 開発環境構築（Python等）
- 8月 開発環境構築（カメラのセット等）
- 9月 開発環境構築（OpenCV等）
プログラムの作成
文化祭の展示
- 10月 プログラムの作成（画像処理）
- 11月 プログラムの作成（顔認識）
- 12月 レポートの作成
- 1月 資料の作成と発表

7 研究成果（近藤）

- ・どんな知識を得たか

Raspberry Pi を通して、現在とても注目されている、「人工知能」、「深層学習」、「Python」の知識を得ることができた。これらの知識は確実に今後、さらに発展していき、さらに身近なものになってくると思われるため、今のうちに知識を得ることができたのはとてもよかった。



図7 顔を認識・出力

- ・プログラム作成の難しさ

普段は使う側の立場であったため、本研究で「作る立場」になれたのは、プログラムの難しさと、数えきれないほどのエラーを体験することができたためだ。それは、自分にとって良い経験となった。

また、0 から深層学習や、Python という新たな言語を学ぶことができ、自分なりに応用するときの振れ幅が広がった。

8 課題

課題としては、Python の知識不足もあり、大半のプログラムが引用して使用していたので、わかる行があればわからない行もあるという、引用するときのデメリットが強くでてしまった。静止面の認識はできたが、動画の認識には成功しなかった。なぜならば、OpenCV の環境に問題があると考えられる。本来 ON にするはずの設定が ON になっておらず、また、ON にする方法がわからず、止まっている現状である。そのため、使用できる機能に制限があり、本研究では特にリアルタイム認識ができないため、今後は設定の操作方法を改めて確認し、問題を解決したい。

また、HTML 文や apache サーバの知識不足があり、簡素なウェブページしか作成できなかった。ウェブカメラで撮影した動画を構築した apache サーバを使ってウェブページに表示させることが二人での最終到達目標であったが、この目標は達成できなかった。

TWE-Lite と Arduino を初めて使い、何も知らず戸惑うことが多かったですがみんなと協力して学んでいくうちにだんだんと知識がついた。

Raspberry Pi によるフィジカル・コンピューティング (AI スピーカーの研究)

研究者：本田 哲也

二見 翔

1 はじめに

研究における達成目標は、2つある。1つ目は、音声での質問に対しての応答である。理由は、文化祭でAIスピーカーの発表を聞き、AIに興味を持ち、私たちはAIスピーカーの研究をしたいと思ったからである。2つ目は、家電制御である。身近な家電を制御し、生活に役立てるものを作りたいと思ったからである。

2 研究の内容

本研究では、Raspberry Pi を使ってどの質問にも答えることができるAIスピーカーを製作し、家電制御をすることにした。APIを利用して開発環境設定をした。その後、身近にある家電の操作をできるように部品を組み立て、設定した。



図1 RaspberryPi3 と VoiceHAT ボード

RaspberryPi と Google AIY Voice kit は先輩が製作したものを再利用した。初期設定では参考文献の記載された OS を使用した。この OS を使用したことで参考文献どおりに進めることができた。また、サンプルコードをダウンロードすることで効率よく研究ができた。



図2 起動画面

3 研究過程

4月～ 7月	AIスピーカーに関する調査
8月～ 9月	Raspberry Pi の設定
10月	AIスピーカー製作用プリント基板への部品実装と動作試験
11月～ 12月	Google AIY Voice Kit の設定 Raspberry Pi の設定と動作試験
1月	家電制御動作実験 レポートの作成、発表準備

4 研究の成果

(1) Wi-Fi・スピーカー・マイクの確認

最初に、ネットワーク接続・スピーカー・マイクの動作確認を行う。「Check Wi-Fi」と「Check audio」をそれぞれクリックすると動作確認を行うことができる。



図3 Wi-Fi とスピーカーの動作確認

(2) API の設定

APIとはソフトウェアやアプリケーションなどの一部を外部に向けて開発することにより、第三者が開発したソフトウェア機能を共有できるようにしてくれるものである。

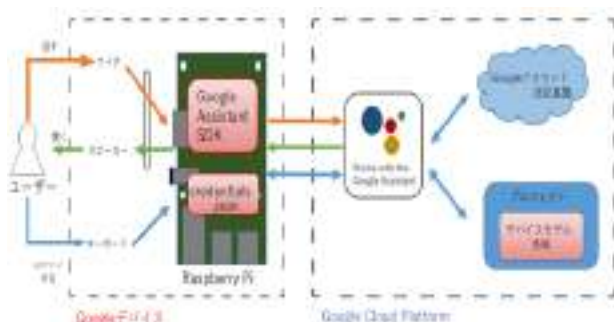


図4 AIスピーカーを自作するための作業概要

(3) AIスピーカー製作用プリント基板とは
トランジスタ技術 2018 3 月号の付録に付いている基板は、Raspberry Pi の拡張ボードである。MEMS マイク 2 個、赤外線リモコン送受信回路、赤外線アレイ・センサ、LED 8 個を搭載している。外部スピーカーを接続し、Raspberry Pi に装着すると AI スピーカーとして動作可能となる。



図5 AIスピーカー製作用プリント基板

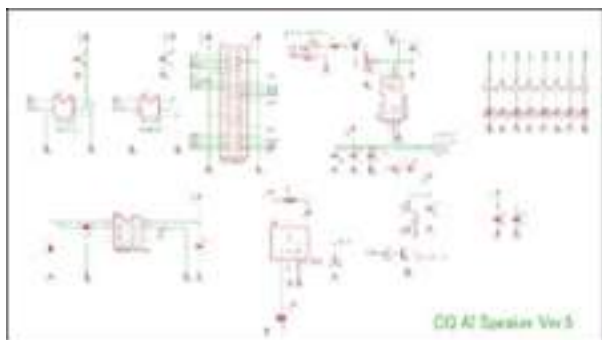


図6 プリント基板の回路図

(4) 家電制御するために行ったこと

ブレッドボードに赤外線LEDを実装してAIスピーカーと接続し、スマホやカメラを使い赤外線LEDが光っているかの確認をした。

参考文献をもとにファイルやコードがあり、それを参考にして作業を進めた。

(5) テレビの操作

RaspberryPi に赤外線LEDを接続、テレビに向け、「src/demo.py」と入力ことでデモプレイが実行される。その後、「Listening…」と表示されるので「テレビをつけて」と言うとテレビがつく。同様に「テレビを消して」と言うとテレビが切れる。「チャンネルをあげて」と言うとチャンネルが1つ変わる。同様に「チャンネルを下げて」と言うとチャンネルが変わる。



図8 テレビ制御の実行画面



←使用した赤外線LED

5 課題・考察

質問に答えることができるAIスピーカーを作成することができた。しかし、完成に至るまでかなり時間がかかってしまった。

課題としては、「OK Google」といえば日本語で質問できるAIスピーカーを目指していたが、「OK Google」と言うと英語で質問や返答も返ってくる。最初は照明の制御をしようとしていたが、照明の制御は本に詳しく書いていなかったが、テレビの制御は詳しく書いてあり、変更をすることにした。変更をする決断が遅く時間がかかってしまった。最終的にテレビを操作することができた。だが、到達目標であった音声での家電製品の制御は結果的に自分たちの理解不足であり、効率よく進めることができなかった。

6 チームの感想

Raspberry Pi というものは何もわからない状態で、様々な専門用語がでてきたため、理解に苦しんだが、インターネットや本で調べたり、先生に聞いたりして、研究を進めてきた。研究の終盤では当初にわからない単語の意味も困ることなく研究を進めることができた。また、製作用プリント基板のはんだ付けをしたことで一から作る楽しさや難しさも学ぶことができた。

課題研究を通して、私たちの持っている知識や技術のレベルを改めて確認することができ、自分の持っている力さらに力をつけることができた。そして、ものづくりの楽しさ苦しさも学ぶことができ、大学、就職につながる様々な体験をすることができた。

スマートフォンを使用した遠隔制御

研究者：驚野 太郎

矢野 海

1 はじめに

私たちは、先輩方が研究を行っていた Arduino を使用した無線通信と近年注目されている XR コンテンツの 1 つである AR について興味をもったため、本研究を行った。

2 研究概要

Arduino と Android で Bluetooth を使用した無線通信を行い、Arduino のセンサで測定したデータを Android に送信する。Arduino のセンサは温湿度センサとダストセンサを使用し、部屋の温度・湿度・埃の量を測定する。

Unity で Android の AR アプリを製作する。AR で魚を表示し、Arduino で測定した部屋の温度によって表示する魚を切り替えられるようにした。

3 研究成果

(1) Arduino

温湿度センサとダストセンサで部屋の温度・湿度・埃の量を測定し、Bluetooth 通信モジュールを使用して、Android にデータを送信する。

Arduino の回路は、図 1 のように組んだ。3 つのセンサの他に Android と正常に通信ができているかを確認するための LED とタクトスイッチも追加した。

プログラムでは、各センサの制御と埃の量の計算や Bluetooth のシリアル通信の設定、Android へのデータの送信を行った。

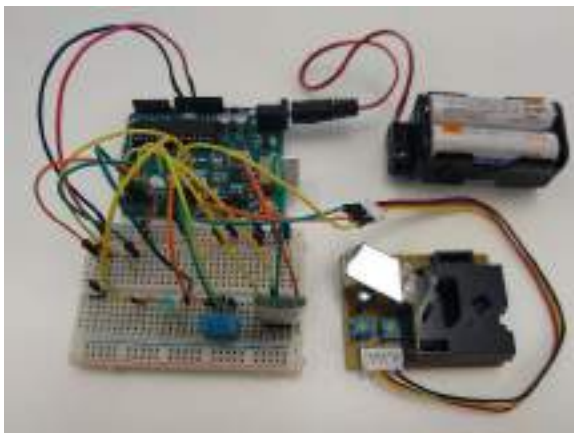


図 1

(2) Android Studio

Android Studio は、Google が提供する Android プラットフォーム向けアプリケーションソフトウェア開発用の統合開発環境である。

Android Studio で行うことは、Unity から呼び出す Arduino とのデータの送受信を行う関数の作成と Android の Bluetooth 機能の使用設定を行い、Arduino と Android の Bluetooth 通信と Unity からの関数の呼び出しに成功し、図 2 のような画面を作成することができた。

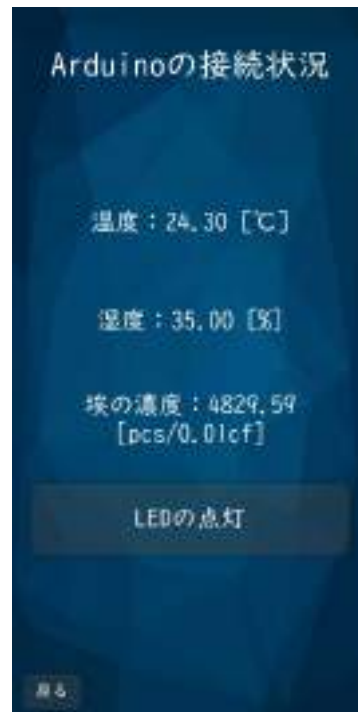


図 2

(3) Unity

Unity は統合開発環境を内蔵するゲームエンジンであり、AR や VR といった XR コンテンツの開発にも対応している。

Unity で行うことは、AR 機能の作成と Arduino のデータをグラフにすることとアプリ全体の UI 等を作成することである。

AR の機能は、平面を検知したらサンゴを表示し、図 3 のように魚を泳がせることである。また、アプリの機能として気温が変わったら表示する魚を切り替えることと、図 4 のようなグラフの作成も行うことができた。



図 3

図 4

(4) Blender

Blender は、3DCG アニメーションを作成するための統合環境アプリケーションである。オープンソースのフリーウェアであり、世界中のユーザーに利用されている。

Blender を用いて、18 匹の魚の 3D モデルを作成した。モデリング、マテリアル、アニメーションを行った。モデリングは、物体の形状、基礎を作ることである。マテリアルは、物体の質感のことでモデリングしたメッシュにマテリアル設定を行うことで質感を表現する。設定項目を適切に設定することで、金属、木材、プラスチック、ガラスなど様々な質感を表現することができる。今回の研究では画像テクスチャをマテリアルとして使用した。様々なアニメーション作成機能が用意されており、機械部品の動作のような変形しないアニメーションや人体のように変形するアニメーションにも対応することができる。ボーンを計 5 個 3D モデルにつけ、滑らかに動かせるよう数値を調整した。



図 5

(5) PicsArt

写真を簡単にアーティスティックな印象に画像加工できるアプリである。

魚の画像を Lowpoly にするために使用した。Blender でも行うことができるが、手間がかかるため、本研究では PicsArt を使用した。

4 まとめ

(1) 成果

Arduino と Android の無線通信に成功し、Arduino のセンサで測定したデータを Android に送信することができた。AR の機能では、うまく魚の表示と切り替えを行うことができた。また、表示範囲を部屋の中に限定し、平面検知によって地面にサンゴを設置することもできた。データをグラフにすることと、Arduino との通信を確認するための画面を作成できた。

(2) 課題

Arduino の回路で、ブレッドボードを使用したことで見栄えが悪くなってしまった。また、同期がうまくできていない時があり、送受信に失敗することもあった。

3D モデルのそれぞれのメッシュの形があまり揃っていない、見た目が悪くなってしまいました。作成段階のときに 3D モデルのサイズを気にしていなく、大きさが不揃いだったので、3D モデルを配置するとき大きさを調整する必要があった。

5 チームの感想

【 鷺野 太郎 】

今回の研究は、複数のソフトウェアを使用したこととハードウェアにも触れたことで多くのことを学ぶことができました。でも、課題も多く残ってしまったのは残念に思います。また、AR と Arduino の無線通信という 2 つのことに挑戦したため、どちらも中途半端になってしまったというのも後悔しています。

【 矢野 海 】

3D モデルを作成するときに、新たな方法を発見できて効率的に作業することができました。もっとさらに効率できる場所があると思うので、ほかの作品などの作成段階を参考にしていきたいです。アニメーションについては今まであまり触れることができなかったため、今回の研究でいろいろなアニメーションやシミュレーションを行ってみて、新たなスキルを習得できてよかったです。

Linux サーバを用いたネットワーク構築

研究者：高橋 海翔 藤田 啓翔

1 はじめに

ネットワーク、データベースについて興味を持ち、学習したいと思ったので、この研究を行った。

2 研究内容

Linux サーバを運用する上で必要不可欠なセキュリティ強化、稼働率の上昇及びデータベースを利用した Web ページの制作

3 研究過程

4月	: 調べ学習
5～7月	: HTML、CSS、PHP、SQLite の学習 サーバ、ルータの設定
8～10月	: Web ページ作成 セキュリティ強化 無線 LAN の設定

4 研究の成果

(1) ネットワーク全体

研究を進めていく中で、以下のネットワークを構築した。

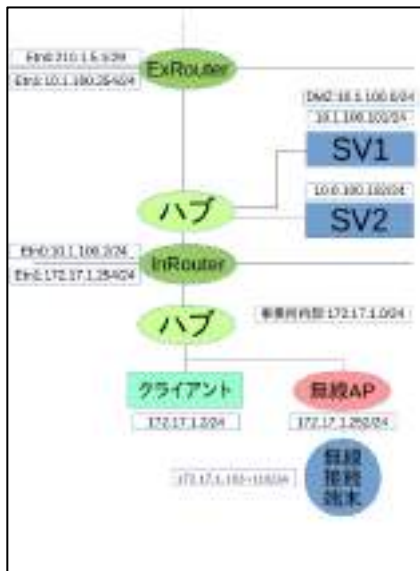


図1 ネットワーク構成図

2 台のサーバは同じサービスを提供している。仮にどちらかのサーバが故障したとしても、もう 1 台のサーバが稼働しているため、2 台のサーバが同時に故障しない限り、サービスを提供し続けることができる。

基本 LAN ケーブルを用い有線で接続しているが、スマートフォンなどポートが存在しない機器は無線アクセスポイントを通して無線で接続する

ことができる。基本的なことだがパスワードを設定し、パスワードを知らないと接続できないようにした。

(2) サーバについて

OS は Debian9.4 stretch を使用した。Debian は有志の団体によってフリーソフトウェアとして配布されており、誰でもダウンロードし利用することができる。

(3) 各種サービス

①DNS (Domain Name System)

ネットワーク上でパケットを送受信するために IP アドレスでコンピュータを特定しなくてはならないが、数字の羅列である IP アドレスは人間にとって覚えにくい。そのため DNS を用いてわかりやすいホスト名に変換して利用する。

DNS は Web サーバやメールの送受信など、多くの場面で利用されるためサーバ設定の最重要項目と言える。

②WWW (World Wide Web)

Web ブラウザを用い、ハイパーリンクで結びつけた世界中のテキスト、ファイルなどにアクセスし閲覧できるインターネットを代表するサービスである。

Apache2 を利用しており、ネットワークに所属している端末から URL「<http://jouhou.young.org>」で接続できる。

③SSH (Secure SHell)

ネットワークを経由し他のコンピュータに接続し、遠隔操作するシステム。接続には公開鍵認証方式を用い、秘密鍵を持たないホストは接続不可能になっている。本ネットワークではクライアントで SV1、SV2 を操作する時に利用している。

(4) Web ページについて

①HTML と CSS

HTML とは、Web ページを作成するための言語で、テキストや画像などを表示させることができる。またリンクを使って他の Web ページに移動させることもできる。HTML はタグを使い記述する。

CSS は Web ページの背景や文字の大きさ、画像の位置など HTML で表示させたものを装飾するのが主な役割。そのため HTML と一緒に使われる。

②PHP (Hypertext Preprocessor)

PHPはプログラミング言語の1つで、HTMLの中にプログラムを埋め込むことができる。HTMLで記述されたWebページをブラウザからの要求に対しサーバが表示させる、「静的なページ」だ。しかし、PHPはブラウザからパラメータとともに送られた要求に対し、サーバがプログラムを呼び出して処理を実行し、パラメータに応じた結果を返す、「動的なページ」である。

③SQL (Structured Query Language)

SQLはデータベースを扱うプログラム言語である。開発当初はSEQUEL(シーケル)と呼ばれており、IBMが開発したものだ。

SQLはリレーショナル・データベース(RDBMS)という種類のデータベースになる。RDBMSではテーブルという表形式の構造でデータを管理するようになっている。Excelと同じで縦列(ROW)を行といい、横列(COLUMN)を列という。そこからSQLを使い、必要なデータを問い合わせ、データを得ることができる。

今回私達が使ったSQLはSQLiteといい、組み込み型の軽量なデータベースソフトである。PHP5.3.0以降に標準で組み込まれており、取扱いが簡単。Android端末にも組み込まれている。

5 まとめ

はじめはサーバ構築についての知識が少なく、基礎知識、設定方法などを調べ、DNS、Web、メール、SSH、DHCPなどの基本的なサービスを提供できた。また、無線アクセスポイントを利用することで、有線LANに非対応の端末は無線LANで接続できるようにした。



図2 無線接続の確認

データベースを用いたサービスを提供するためアンケートページを作成した。HTML、CSSを用いたWebページの作り方、PHPを用いたデータの送信、SQLを用いたデータの収集について学び、Webページとして実装することができた。

収集したデータは確認用のページで表・グラフとして閲覧することができる。



図3 アンケートページ・データベースの確認

2台のサーバの連携強化、IPv6の実装、クライアントサーバスクリプトを用いたWebページのリアルタイム更新など、課題は多く残っている。機会があれば調べ、実装できるようにしたい。

6 チームの感想

【 高橋 海翔 】

初めてのことでばかりで戸惑いでしたが、本やインターネット上の文献などを利用し、少しずつ理解し進めていくことができました。文化祭までに形にすることができてよかったです。しかし、課題も多く、まだ改善できることがあったと思います。機会があれば改善させたいです。

【 藤田 啓翔 】

今回の研究の中で、PHPとSQLのプログラムを組み立てるのに苦労しました。PHPはC言語に似ていると感じ、驚きました。アンケートページと結果のページを作る中でPHPはHTMLやCSSよりも難しく感じ、苦労しました。SQLは今まで見たことがないタイプのプログラム言語だったため、理解するのが大変でした。まだできることはあったけれど、本来の目的は達成できたので良かったです。

情報技術科 HP の改善

研究者：仙石 幸久

1 はじめに

近年、ICT 社会・ソサエティ 5.0 というワードが社会的に浸透し、今やパソコンなしには生活できないレベルになりつつある。そういった現代の社会において、学校の宣伝や紹介を、画面を通して発信していくものがホームページである。

本校のホームページ内には、情報技術科の学科ホームページがある。この学科ホームページを、より分かりやすく正確に外部の方たちへ発信するために、改善していくことができると考え、本研究に着手することにした。

2 研究の内容

本研究では情報技術科 HP のデザインやアニメーションに、改善できる点が複数あることに着目し、ホームページ編集の基礎となる HTML と CSS の知識・技術を習得することから始めた。

今回は主にパララックス・ファビコンについて改善を行った。

改善するためのコーディングには Windows に標準装備されている、「メモ帳」を使用した。様々なテキストファイルを扱うにあたって、とても便利なソフトである。



```
<!doctype html>
<html lang="ja">
  <head>
    <meta charset="utf-8">
    <title>タイトル</title>
    <style>
      h1
        color:#0000FF;
    </style>
  </head>
  <body>
    <h1>見出しを表示</h1>
    <p>コメント</p>
    <p>本文を表示</p>
  </body>
</html>
```

図 1 HTML と CSS を使ったコード例

3 研究過程

- 4～5月 HTML と CSS の学習
- 6～9月 Bootstrap、jQuery、アニメーションについての勉強、HP のデザイン案の作成
- 10月 HP の作成
- 11月 文化祭展示ポスター作成
- 12月 レポートの作成
- 1月 発表準備

4 研究の成果

(1) アニメーションの追加

アニメーションは、ユーザーの操作によって開始するものを多く取り入れた。このようにすることで、文章を読む際に注意がそれることがないようにしている。

追加したアニメーションはすべて、style タグ内に書いた。追加したアニメーションは、フェードイン、アンダーラインである。



```
vertical-align: middle;
border-bottom: solid 1px #0000;

.fade-in {
  opacity: 0;
  transition: opacity 0.5s;
  transition: all 0.5s;
  display: block;
  margin: 0 auto;
}

.fade-in:hover {
  opacity: 1;
  transition: 1 translate(0, 0);
}

.fade-in {
  opacity: 1;
}

<!--HEROET FOR-->
<div id="header">
  <div class="container">
    <div class="row">
```

図 2 フェードインの CSS 設定

(2) デザインの変更

元々のデザインは、シンプルであると同時に殺風景な印象を与えていた。



図3 現在の学科ホームページ

変更点として画面上部に写真と文字を入れた。この写真には、薄い黒のフィルターをかけることで写真中央に入れた文字が見やすくなるようにしている。

また、フォントの変更を行った。フォントの変更は、表題にだけ適用した。これにより、本文の可読性を失うことはない。



図4 改善した学科ホームページ

5 考察

本研究では、Bootstrapを使用してホームページを作成している。これにより、様々な拡張機能を使用することができるようになった。

しかし、その拡張機能を使用することができなかった。理由として、BootstrapのCSSファイルを読み込んでいなかったからだと考えられる。Bootstrapの読み込み方法は、インターネットからCDNを読み込む方法と、あらかじめCSSファイルをダウンロードしておく、CSSファイルを読み込む方法がある。最初はCDNを使用した方法を使おうとしたが、適用することができず、ファイルから読み込むことにした。そうしたところ、適用することができた。

6 まとめ

本研究から、ホームページを作成する技術を学んだ。そして、今まで見てきたホームページはこのように作られているのだと感じ、アニメーションなどを多く使用したホームページを作成するのはとても難しいということを知った。

研究を進めていくうちに、ホームページを作成するためのオープンソースソフトウェアが他のプログラミング言語よりも多く感じた。本研究ができているのはオープンソースソフトウェアのおかげであるため、オープンソースソフトウェアを提供されている人たちには感謝しなければならない。