

平成 24 年度

学士学位論文

LED マトリクスを用いたアート作品 のためのビジュアルプログラミング環境

A visual programming environment
for art work using a LED matrix

1130302 今井 達朗

指導教員 高田 喜朗

2013 年 3 月 1 日

高知工科大学 情報学群

要 旨

LED マトリクスを用いたアート作品 のためのビジュアルプログラミング環境

今井 達朗

近年, マウスやディスプレイ, キーボードなどの入出力装置だけではなく, センサやアクチュエータなどを使って人間とコンピュータとの情報のやりとりを実現するフィジカルコンピューティングが注目されている. フィジカルコンピューティング開発を行うには電子回路やプログラムの知識が不可欠であり, それらの知識のない人たちにとって製品の開発を行うことは難しい. フィジカルコンピューティング開発を簡単に行うツールとして Arduino がある. Arduino は, マイクロコントローラ, 入出力ポートを備えた基板, それを制御するソフトウェアの開発環境により構成されている. しかし, Arduino は電子回路を組み立てる手間を大きく削減しているが, プログラミング能力は不可欠である.

本研究では, 既存研究で扱われなかつた対象に対して適切なプログラミング環境を考察することを目的とし, 具体例として LED マトリクスを用いたアート作品を対象とするプログラミング環境について検討, 実装する.

キーワード フィジカルコンピューティング, Arduino, LED マトリクス, ビジュアルプログラミング言語

Abstract

A visual programming environment for art work using a LED matrix

Tatsuro IMAI

In recent years, *physical computing*, which realizes exchanging information between people and computers by using not only a mouse, a display, and a keyboard but also I/O devices such as a sensor, an actuator, and so on, is gathering attention. The knowledge of the electronic circuit and programming is indispensable to perform physical computing development, and it is difficult to develop the product for people without those knowledge. Arduino is one of the tools that enable a user easily perform the physical computing development. Arduino consists of a microcontroller, a board with the I/O ports, and development environment of software to control it.

Although Arduino is greatly diminishing time to compose electronic circuit, programming skill is still indispensable. In this study, we design a visual programming environment for art work using a LED matrix and then implement it.

key words physical computing, Arduino, LED matrix, visual programming langage

目次

第 1 章 序論	1
1.1 研究目的とその背景	1
1.2 本稿の構成	2
第 2 章 背景知識	3
2.1 フィジカルコンピューティング	3
2.2 Arduino	4
2.3 ビジュアルプログラミング言語	5
2.4 Intuino	7
第 3 章 提案システム	9
3.1 提案システムのインターフェース	9
3.2 利用者の操作方法	10
第 4 章 評価	17
4.1 実験	17
4.1.1 実験内容	17
4.1.2 実験結果	19
第 5 章 まとめと今後の課題	20
謝辞	21
参考文献	22

図目次

2.1	Haohao_table[5]	3
2.2	Arduino	4
2.3	ArduinoIDE	5
2.4	Scratch[7]	6
2.5	ToonTalk[8]	6
2.6	Intuino[3]	8
3.1	提案システムの外観	10
3.2	プログラム例	11
3.3	初期状態	12
3.4	点灯させたいコマの作成	13
3.5	コマの追加	14
3.6	次のコマの作成	15
3.7	2コマ目の追加	16
4.1	LED マトリクス	18

表目次

第1章

序論

1.1 研究目的とその背景

近年, マウスやディスプレイ, キーボードなどの入出力装置だけではなく, センサやアクチュエータなどを使って人間とコンピュータとの情報のやりとりを実現するフィジカルコンピューティング [1] が注目されている. フィジカルコンピューティング開発を行うには電子回路やプログラムの知識が不可欠であり, デザインやアートを仕事とする人たちにとって製品の開発を行うことは難しい. フィジカルコンピューティング開発を簡単に行うツールとして近年 Arduino[2] が注目を集めている. Arduino は, マイクロコントローラ, 入出力ポートを備えた基板, それを制御するソフトウェアの開発環境 ArduinoIDE により構成されている.しかし, Arduino は電子回路を組み立てる手間を大きく削減しているが, プログラミング能力は不可欠である. それを解決すべく, 提案システムではテキスト形式のプログラミング言語ではなくビジュアルプログラミング言語を採用する. フィジカルコンピューティング開発を支援する既存の研究として Intuino[3] がある. Intuino は LED やモータなどの動きをタイムラインにより作成する機能, ビジュアルプログラミング言語により入出力デバイスとの接続を記述する機能などを備えており, センサやアクチュエータの動きを視覚的にデザインすることができる. Intuino により, LED 等の单一出力の時間変化を視覚的にプログラミングすることができるが, LED マトリクスのような, 複数の出力が空間的に配置された装置を対象としている場合は扱うことができない.

本研究では, 既存研究で扱われなかつた対象に対して適切なプログラミング環境を考察することを目的とし, 具体例として, LED マトリクスを用いたアート作品を対象とするプログ

1.2 本稿の構成

ラミング環境について検討, 実装する.

1.2 本稿の構成

本論文は, 本章を含め 5 章で構成される. 第 2 章では, 背景知識について述べる. 第 3 章では, 提案システムについて述べる. 第 4 章では, 評価について述べる. 第 5 章では, 結論を述べ, 今後の課題を示す.

第 2 章

背景知識

2.1 フィジカルコンピューティング

マウスやキーボード、ディスプレイといった標準的な入出力デバイスではなく、センサやモータなどを使い情報のやりとりを実現することをフィジカルコンピューティングという。フィジカルコンピューティングの例として、任天堂の wii リモコン [4] が挙げられる。wii リモコンは、今までのボタンの操作だけのコントローラとは違い、加速度センサとジャイロセンサが搭載されており、wii リモコンを振り回したり、画面に向けたりすることでゲームを楽しむことができる。図 2.1 は Haohao_table[5] とよばれる作品である。テーブルの中心にあるプランタには距離センサが六方向に内蔵しており、人やテーブル上のモノの距離によって様々な模様が描かれる。



図 2.1 Haohao_table[5]

2.2 Arduino

これらのように、最近では従来のキーボードやマウスを使ったコンピュータの操作から抜け出して、自由に情報のやりとりを楽しむフィジカルコンピューティングが日々研究されている。フィジカルコンピューティングの構成要素は光、音、温度、加速度などのセンサやモータ、ソレノイドなどのアクチュエータとマイコンである。フィジカルコンピューティング開発を行う場合、電子回路やプログラムといった、ハードウェアとソフトウェアの両方の知識を含めて考える必要がある。

2.2 Arduino

Arduino は、マイクロコントローラ（図 2.2）、入出力ポートを備えた基板、それを制御する ArduinoIDE と呼ばれるソフトウェアの開発環境により構成されている。Arduino はエンジニア以外にもわかりやすいように説明や開発環境が工夫されており、簡単にマイコンを制御することができる。モジュール化された入出力機能とその扱いやすさから、フィジカルコンピューティングを考えるためのツールとして、教育の一環やデザインやアートを仕事とする人の間で広く普及している。

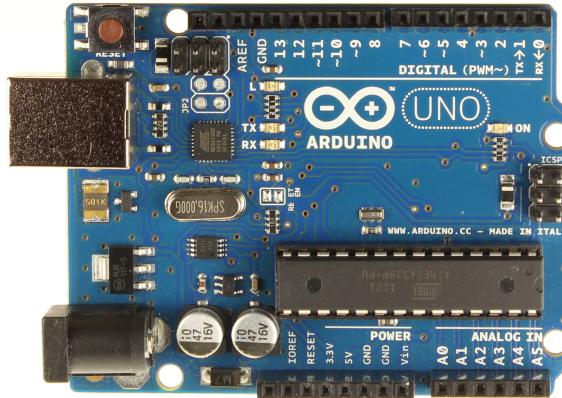


図 2.2 Arduino

ArduinoIDE（図 2.3）は C++に似たプログラム言語で開発を行う。記述言語が高い自由度を持っているため、様々な処理を行うプログラムを記述することができる。

2.3 ビジュアルプログラミング言語

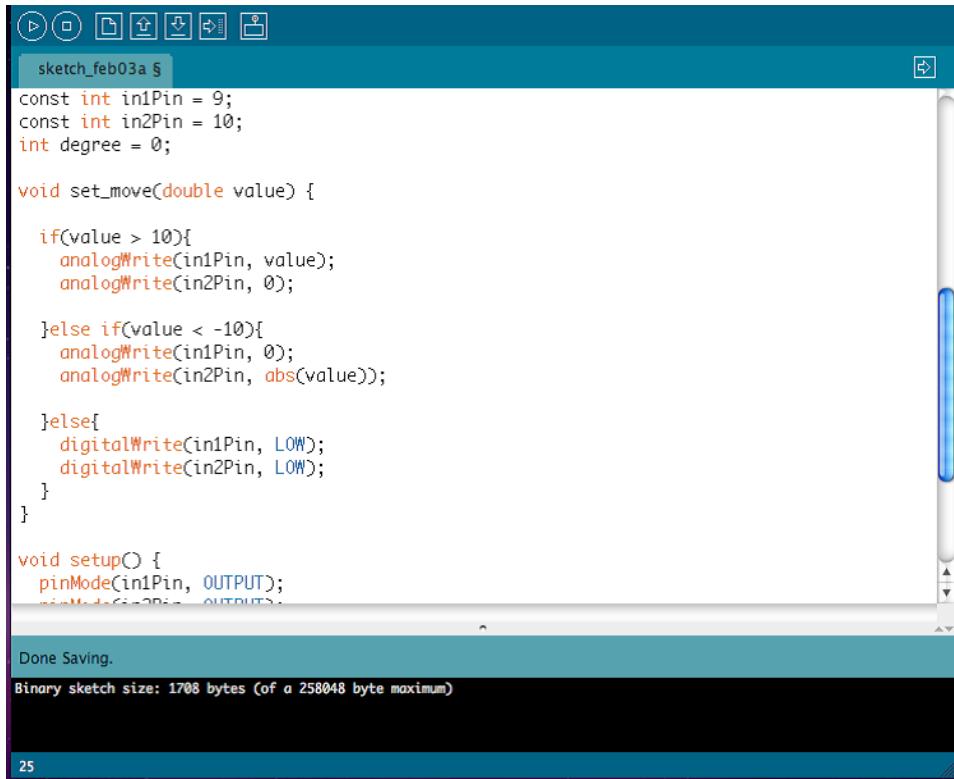


図 2.3 ArduinoIDE

2.3 ビジュアルプログラミング言語

ビジュアルプログラミング言語 [6] とはプログラムの構造をテキスト形式で表すのではなく、絵やアイコンを含んだり、関連を線で示すなどグラフィカルに操作することでプログラムを作成する方式のプログラミング言語である。既存のビジュアルプログラミング言語として Scratch[7] (図 2.4) が挙げられる。これは、MIT Media Lab の Lifelong Kindergarten Group が開発した子供向け教育用プログラミング環境である。レゴブロックのようなタイルを積み重ねることで、スプライト (画像オブジェクト) を動かすスクリプトを作成する。

また、子供向け教育用プログラミング言語として ToonTalk[8] (図 2.5) がある。変数は「箱」、手続きは「ロボット」というようにプログラムの各要素がグラフィカルなオブジェクトに置き換えられており、マウスを利用して直感的にゲーム感覚でプログラムを作成できる。ToonTalkにおいて、「プログラムを記述して実行する」ということは「ロボットを訓練して、実行する」ということに対応する。

2.3 ビジュアルプログラミング言語

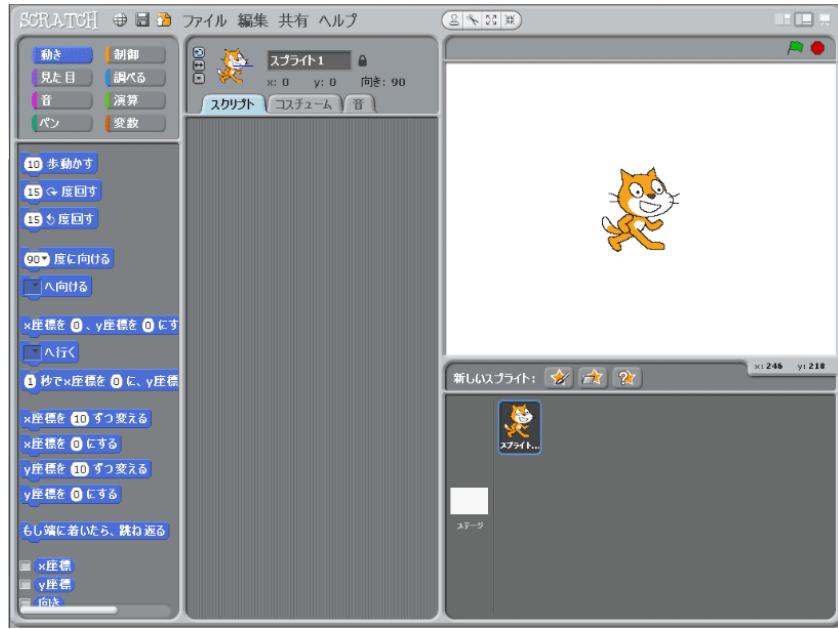


図 2.4 Scratch[7]



図 2.5 ToonTalk[8]

これらのようにゲーム感覚でアルゴリズムやプログラムの処理の流れなどを理解させるために、子供向け教育用にビジュアルプログラミング言語が用いられることがある。

また、ビジュアルプログラミング言語は教育用以外にも使用される。ビジュアルプログラ

2.4 Intuino

ミング言語を用いることにより、プログラミングの知識が少ない人でも容易にプログラミングが可能となる。

2.4 Intuino

フィジカルコンピューティング開発を支援する既存研究として Intuino (図 2.6) がある。Intuino は、センサ及びアクチュエータの組み合わせとその詳細な挙動を、プログラミングすることなしに視覚的にデザインできるツールである。Intuino はいくつかの機能があり、それらの機能について説明していく。

■操作対象の状態を視覚的に操作 センサやアクチュエータはイラストによって表現され、各部品をクリックすることで各プロパティが表示される。各部品に合わせてプロパティを設定していくことで、各部品の機能を理解しながら操作することが可能となり、基本操作に対するミスを軽減できる。アクチュエータの動きはタイムラインで調整することができる。

■動きをタイムラインで制御 タイムラインを用いることによって、アクチュエータの動きを編集できる。タイムライン上にベジュ曲線を描写し、時間軸にそって値の大きさを操作することで、滑らかな動きや細かな動きを作成することができる。

■センサの値のグラフ描写 Intuino と Arduino の I/O モジュールは常に連動しており、Arduino に接続されているセンサの値がリアルタイムに画面に反映される。時間軸に沿ってセンサの値が左へ流れながら蓄積されることで、センサ値の変わりをグラフィカルに表示することができる。

■イベントの組み合わせ センサの値とアクチュエータのタイムラインをイベント用のラインで結ぶことで、センサの値に応じたアクチュエータの動きなどができる。例えば、人が近づくと LED が光るといった処理を GUI 上で組み立てることが可能になる。

2.4 Intuino

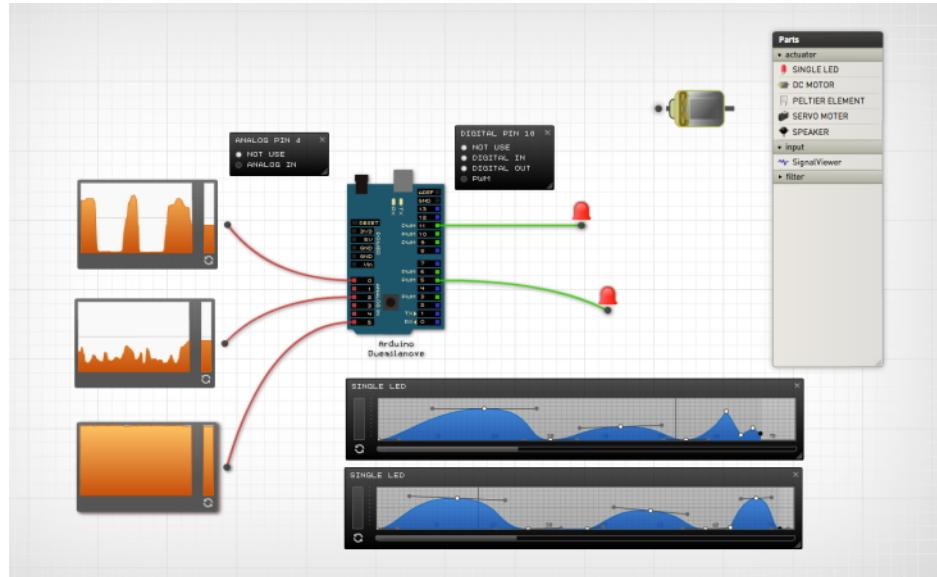


図 2.6 Intuino[3]

第3章

提案システム

Intuino により, LED 等の单一出力の時間変化を視覚的にプログラミングすることができるが, LED マトリクスのような, 複数の出力が空間的に配置された装置を対象としている場合は扱うことができない.

本研究では, 既存研究で扱われなかつた対象に対して適切なプログラミング環境を考察することを目的とし, 具体例として, LED マトリクスを用いたアート作品を対象とするプログラミング環境について検討, 実装する.

3.1 提案システムのインターフェース

LED マトリクスは, 複数の LED をマトリクス状に配置し, その明滅により文字や絵を表現する装置である. LED マトリクスを用いた作品をプログラミングの知識のない利用者でも簡単な操作でプログラミングができるシステムを提案する. そのようなシステムには以下の機能が必要と考える.

- プログラミングの知識を必要としない.
- 視覚的にプログラミングできる.
- 複数の出力をそれぞれ操作できる.
- システム上でセンサやアクチュエータの動きを確認できる.

本システムではビジュアルプログラミング言語を用いることにより, 利用者がタイピングを必要とせず, プログラムを作成できるようにする. 図 3.1 は提案システムの外観を示す. 提案システムではアニメーションのように一つ一つのコマを作成し, 作成した複数のコマを連

3.2 利用者の操作方法

続して表示させる。一コマ毎にコマの切り替わる速さを調節できる機能を備えており、より自由な作品を作り上げることができる。作成した複数のコマは、提案システム上で Arduino で動作する言語のソースコードに変換することができる。また、実際に LED マトリクスを光らせずに、提案システム上でどのように光るのか確認できる。

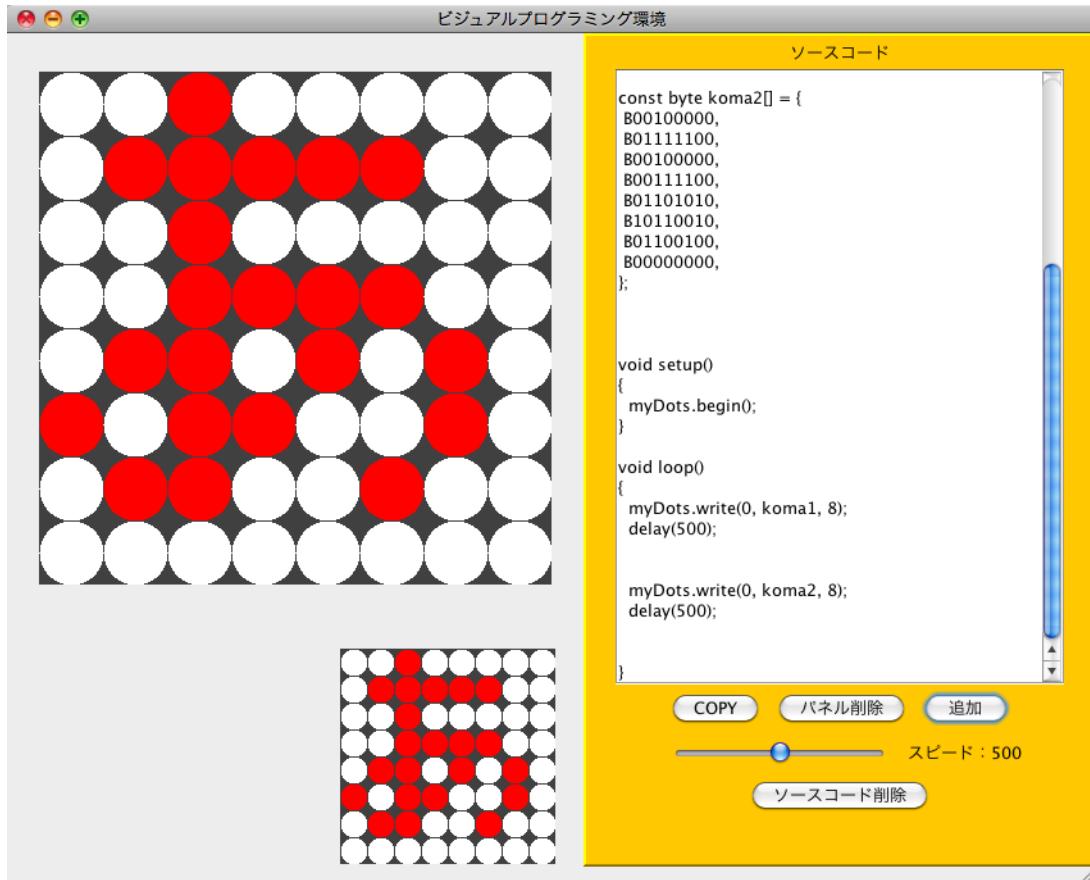


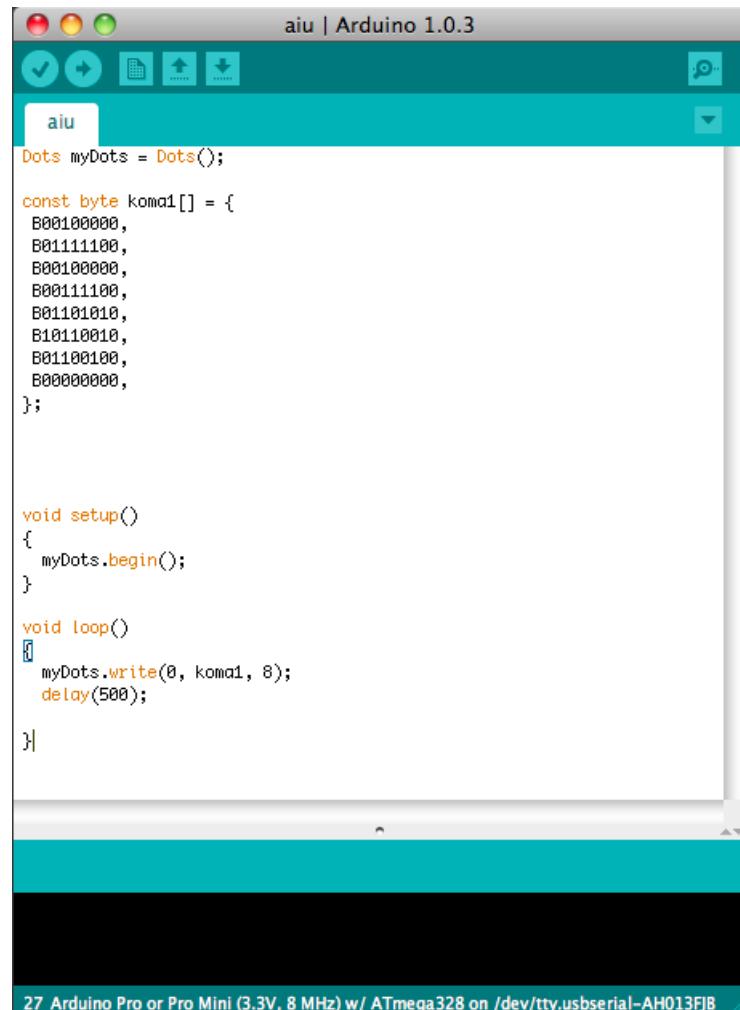
図 3.1 提案システムの外観

3.2 利用者の操作方法

図 3.2 は ArduinoIDE を用いて LED マトリクスを「あ」と光らせるプログラム例である。アニメーションの様に次のコマを表示するには、次のコマを光らせるプログラムを追加して書く必要がある。また、プログラムだけでは、LED マトリクスが実際どのように光るか分からづらい。そのため、実際に LED マトリクスを動作させて確認しながら、理想とする動きを

3.2 利用者の操作方法

作成していく必要がある。



The screenshot shows the Arduino IDE interface with a sketch titled "aiu". The code is as follows:

```
Dots myDots = Dots();

const byte komai[] = {
  B00100000,
  B01111100,
  B00100000,
  B00111100,
  B01101010,
  B10110010,
  B01100100,
  B00000000,
};

void setup()
{
  myDots.begin();
}

void loop()
{
  myDots.write(0, komai, 8);
  delay(500);
}
```

The status bar at the bottom indicates: 27 Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328 on /dev/tty.usbserial-AH013FJB.

図 3.2 プログラム例

3.2 利用者の操作方法

提案システムでは、LED マトリクスがどのように光るか提案システム上で確認しながらプログラムを作成できる。次に提案システムの操作方法を説明する。操作方法は以下の通りである。

1. 提案システムの画面の左側にある 8×8 のマスは LED マトリクスを表す。このパネルを使い目的の絵を描いていく。図 3.3 は提案システムの初期状態である。

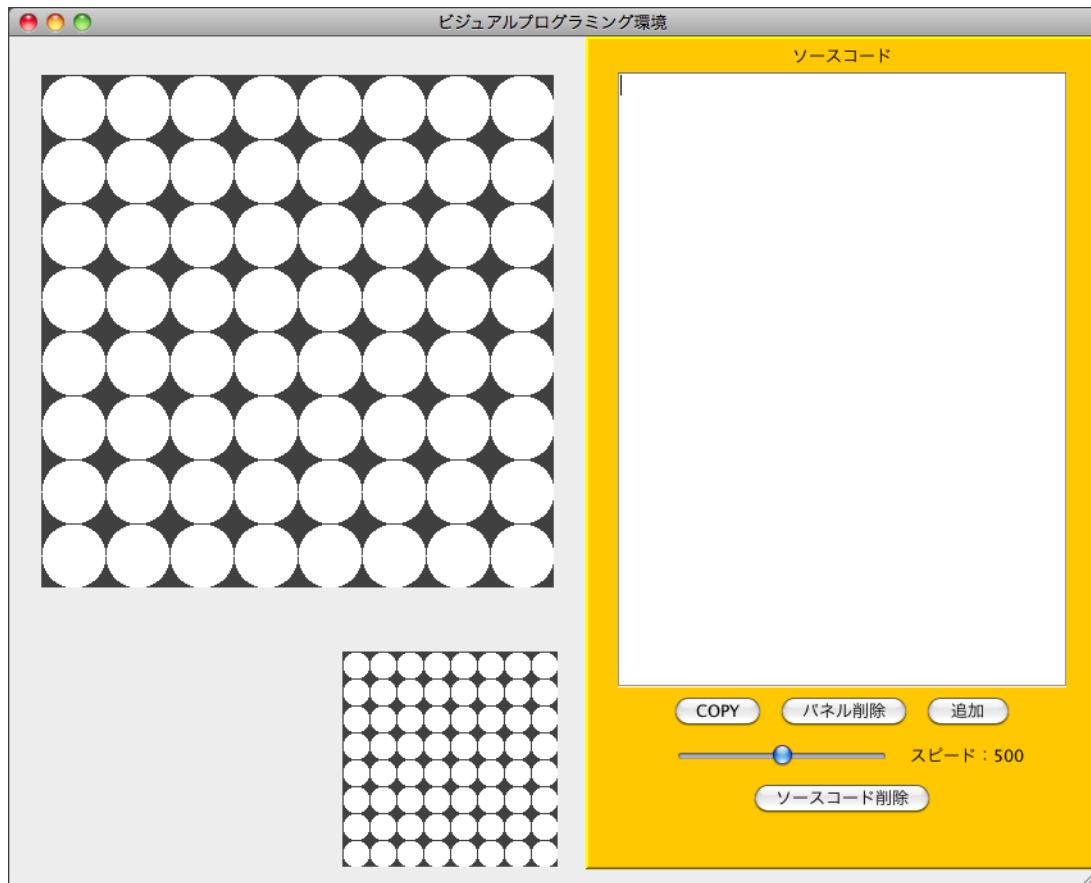


図 3.3 初期状態

3.2 利用者の操作方法

2. パネルの各マスはマウスの左クリックで明滅を切り替えることができる。白色のマスは LED が光っていない状態を表す。赤色のマスは LED が光っている状態を表す（図 3.4）。

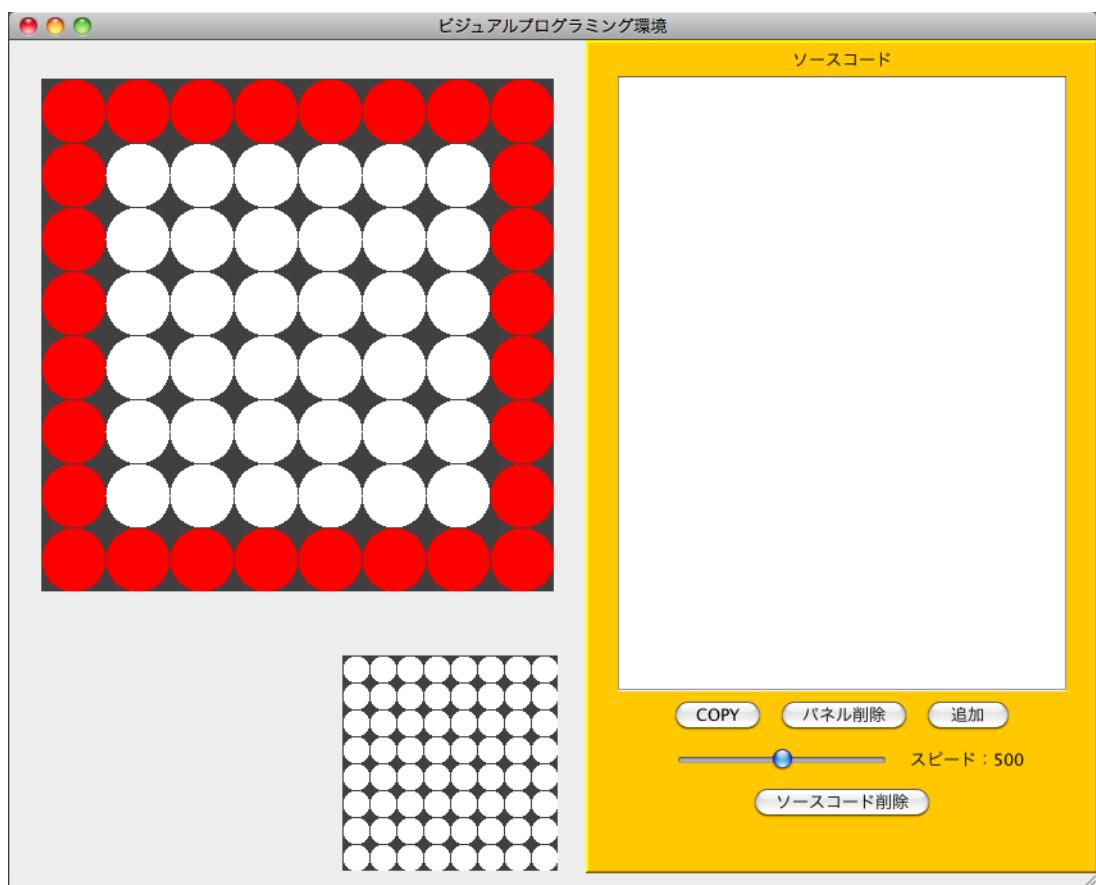


図 3.4 点灯させたいコマの作成

3.2 利用者の操作方法

3. 目的とするコマを作成した後, 追加ボタンを押し, コマを追加する. すると, 右側のパネルに Arduino 上で動作するソースコードが生成される. また, 左側のパネルでは, 赤色の点灯するマスは薄い赤色に切り替わる. これは, 次のコマを作成する上で便利なよう以前追加したコマを見やすくするためである. 薄い赤色のマスは LED では光っていない状態を表す. 追加ボタンの下のスライダーは, 追加するコマが表示される時間 (ミリ秒) を表す. 追加ボタンを押す前にスライダーを調節することで, コマが切り替わるスピードを変更することができる (図 3.5) .

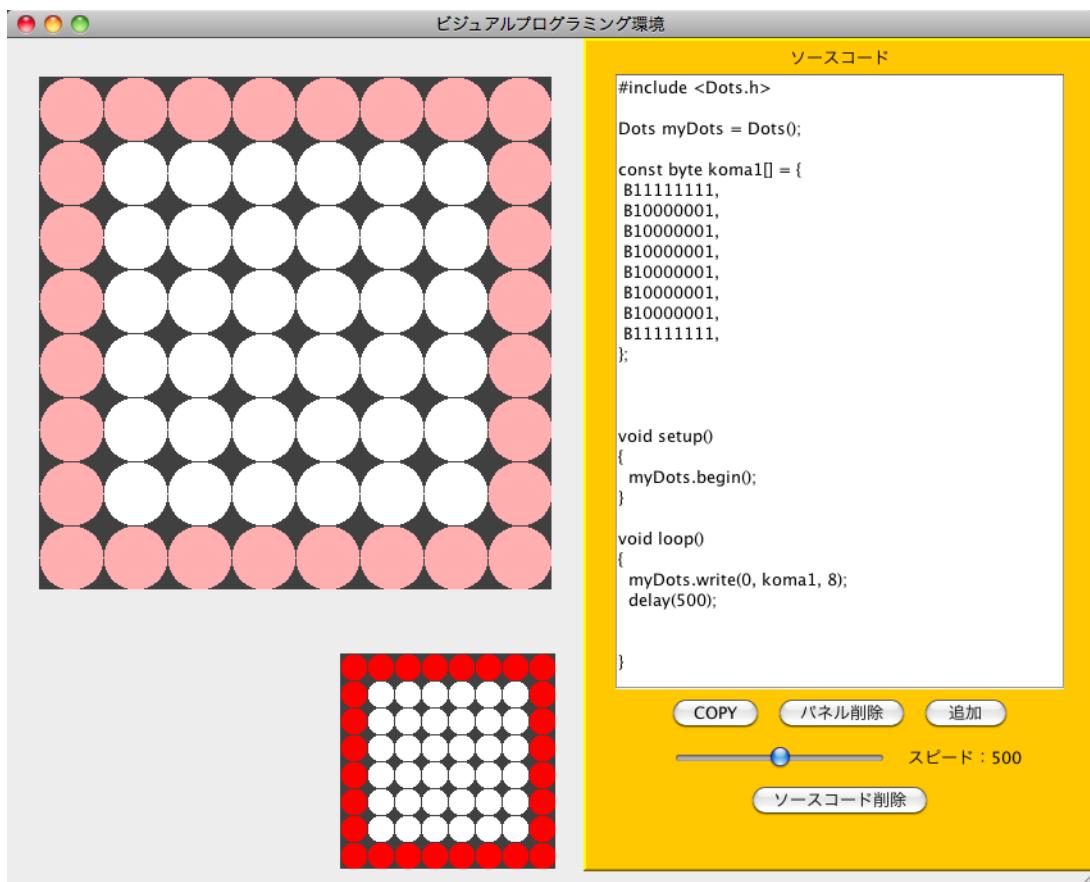


図 3.5 コマの追加

3.2 利用者の操作方法

4. 同様に次のコマを作成していく（図 3.6）。

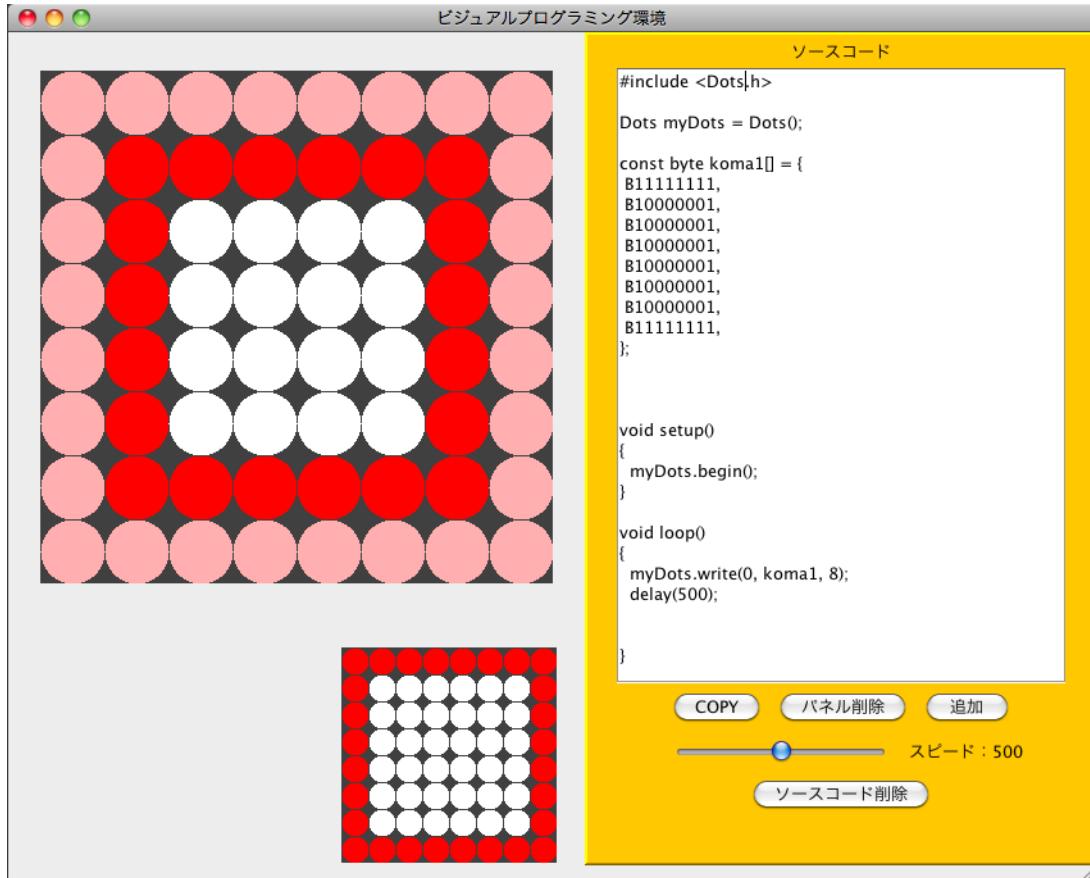


図 3.6 次のコマの作成

3.2 利用者の操作方法

5. 同様に二枚目以降のコマを追加すると, Arduino のソースコードも自動的に変更される
(図 3.7) .

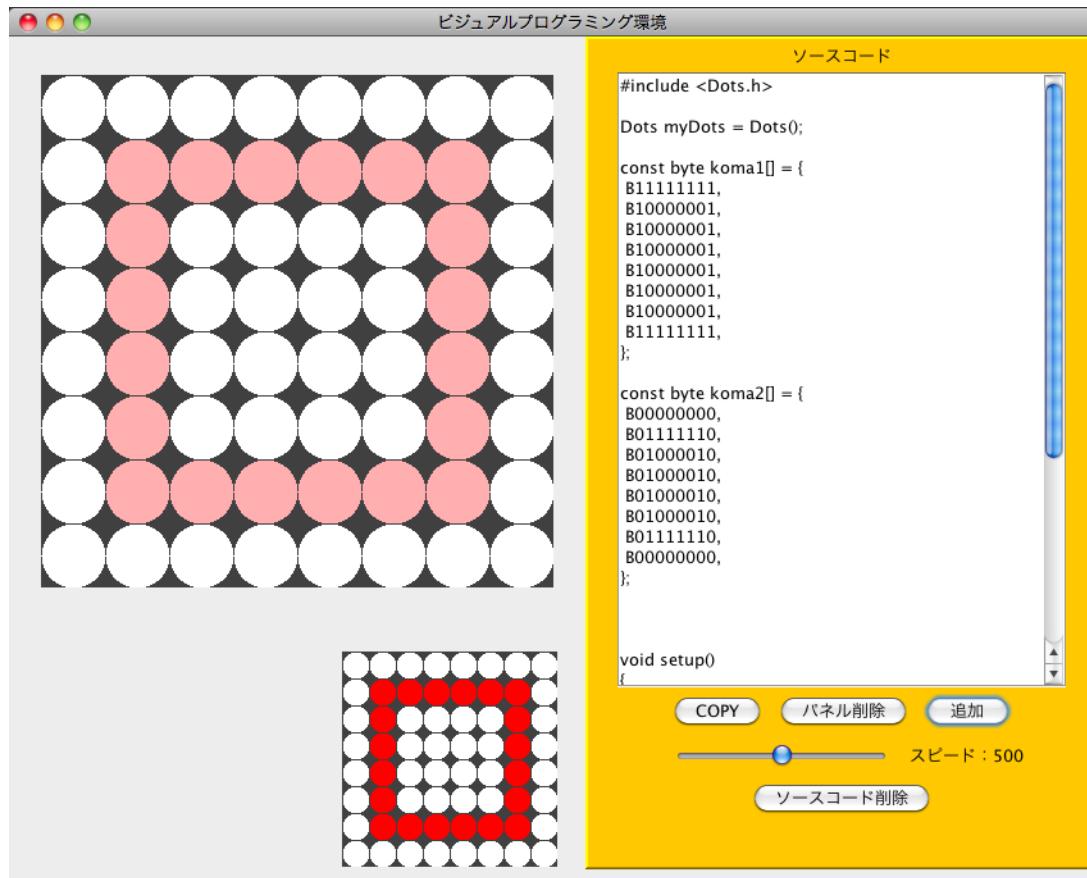


図 3.7 2 コマ目の追加

第 4 章

評価

4.1 実験

4.1.1 実験内容

予備実験としてプログラミング知識のある本学情報学群 4 年生の被験者 2 名に提案システムおよび ArduinoIDE を用いて、LED マトリクスを「あ、 い、 う」の 3 文字が 0.5 秒毎に切り替わるプログラムを作成してもらった。図 4.1 は、作成してもらったプログラムによる LED マトリクスの動作の様子である。Arduino 言語で動作する LED マトリクスハードウェアは、あらかじめこちらが用意し、被験者はプログラミングのみを行うものとする。実験の目的は ArduinoIDE よりも提案システムの方が容易にプログラム作成が行えること、ならびに提案システムはごく簡単な説明だけで操作できることを確認することである。

4.1 実験

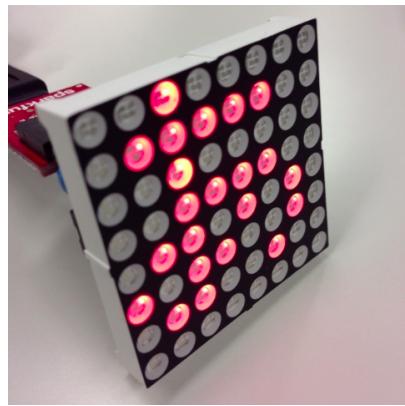


図 4.1 LED マトリクス

4.1 実験

4.1.2 実験結果

両被験者とも ArduinoIDE では完成するまで 30 分程度かかったのに対し、提案システムでは 5 分程度でプログラムを完成させることができた。ArduinoIDE では、度々 LED マトリクスを光らせて動作を確認を行ったり、予想していた箇所が光らないなどで時間を費やしていた。被験者の感想として、提案システムは実際に LED マトリクスを光らせずに、提案システム上でどのように光るのか確認できるため使いやすいとのコメントを得た。提案システムはプログラミングの知識のある利用者にとっても ArduinoIDE より容易にプログラミングを行うことが可能であること、ならびに提案システムは事前の説明はほとんど要さず直感的にプログラミングが可能であることを確認できた。

第 5 章

まとめと今後の課題

本研究では、既存研究で扱われなかつた対象に対して適切なプログラミング環境を考察を行い、LED マトリクスを用いたアート作品のためのビジュアルプログラミング環境を構築した。予備実験により、提案システムはプログラミング経験者にとって ArduinoIDE より容易にプログラミングが可能であること、ならびに提案システムは事前の説明はほとんど要さず直感的にプログラミングが可能であることを確認できた。

今後の課題として、プログラミングの知識のない被験者で同様の評価実験を行い、プログラミングの知識がなくても提案システムを使用することにより容易にプログラムが作成できることを確認する予定である。また、今回の提案システムは LED マトリクスを対象としたシステムを提案したが、LED マトリクス以外のアクチュエータでも利用できるようさらなる機能追加や、LED のオンオフだけでなく、ゆっくり LED が光ったり、波打つような模様が表示されるといった、より複雑に LED マトリクスを光らせるシステムが望まれる。

謝辞

本研究を進めるにあたり、御指導、御鞭撻を賜りました主指導教員である本学情報学群高田喜朗准教授に心より深く感謝致します。また、副査として貴重な御助言を頂きました任向實教授ならびに酒居敬一講師に心より感謝致します。

同研究室修士2年の山崎康生氏には日頃から多大な御指導、ご助言をいただき、心より感謝いたします。

本研究の被験者実験に参加してくださった方々には貴重な時間を割いていただき、厚く感謝いたします。

最後に、今までの学生生活を身近で支えてくださった同期の学友の皆様と高知工科大学の教職員の皆様に感謝いたします。

参考文献

- [1] D. O'Sullivan and T. Igoe, Physical Computing: Sensing and Controlling the Physical World with Computers, Course Technology, 2004.
- [2] Arduino Software, "Arduino," <http://www.arduino.cc/>
- [3] 姉崎 祐樹, 脇田 玲, "Intuino: ビジュアルプログラミングによるフィジカルコンピューティング開発支援環境の構築," 情報処理学会研究報告, 2010-HCI-137, pp.1–4, 2010.
- [4] 任天堂 wii, <http://www.nintendo.co.jp/wii/>
- [5] 小林 茂, "メディアアート紀行 (No.8) ツールキット Gainer をめぐる旅," 映像情報メディア学会誌, Vol.64, No.11, pp.1598–1601, 2010.
- [6] P. T. Cox, "Visual Programming Languages," Wiley Encyclopedia of Computer Science and Engineering, Vol.5, pp.3015–3025, 2009.
- [7] Scratch, <http://Scratch.mit.edu/>
- [8] ToonTalk, <http://www.toontalk.com/>