

Arduino を用いた会話支援システムの試作

平井佑樹^{†1} 佐山裕一^{†1} 三澤大地^{†1} 金子敬一^{†1}

近年の学校教育では学習者の学習能力を高めるために、学習者主体の学習方法が重視されるようになってきている。そのなかで学習者が数人程度のグループを組み、自ら課題を発見し互いに助け合って問題解決していく協調的な学習方法が有効とされている。このような協調学習では学習者同士が対話によって、自らの学習知識を外化し、体系化していく知識構築の過程が重要である。しかし、3 人以上のグループにおける対話では、「ある 1 人がずっと話し続けている」、「特定の 2 人だけが対話していて、それ以外の人はそれをずっと聞いているだけになる」ことがある。このような状況では、自らの学習知識を外化することができない人がでてきてしまう。

そこで本研究では、協調学習の参加者全員が均等に知識を外化できるように支援するシステムを、Arduino を用いて開発した。このシステムは、「5 秒以上発話をしない参加者」、「20 秒以上発話をしない参加者」、「10 秒以上発話を続けている参加者」を支援対象と判断し、システムに接続されている LED ランプを光らせることによって、参加者に支援対象であることを知らせる。システム利用実験の結果、実験参加者から、「誰が喋っているか、いないか」を知ることが出来ることが良いという意見が得られた。

Trial Development of a Conversation Support System using Arduino

YUKI HIRAI^{†1} YUICHI SAYAMA^{†1}
DAICHI MISAWA^{†1} KEIICHI KANEKO^{†1}

Several hundred researches mentioned that student-centered learning is important for enhancing students' learning skills in recent school education. Collaborative learning, which is the learning process in which two or more participants attempt to study something together, is also effective for problem-solving. It is important for participants to externalize their opinions and reorganize their knowledge by communicating each other in the collaborative learning. However, in three-party face-to-face collaborative learning, there are situations such as 'A participant keeps speaking for a long time' or 'Specific two participants keep talking each other for a long time and other participant keeps listening to the talking'. In these situations, specific participant(s) cannot externalize their opinions and/or knowledge.

In this study, we have developed a conversation support system that participants can equally externalize their opinions and/or knowledge. We have implemented the system by using Arduino. The system identifies the targets such as 'A participant does not speak for more 5 seconds', 'A participant does not speak for more 20 seconds', or 'A participant keeps speaking for more 10 seconds', and supports such targets by turning on the LED bulbs. As a result of the pilot experiment, participants could clearly identify who is speaking for a long time or not.

1. はじめに

仲間とともに問題解決や学習を行う協調学習は、近年の学校教育で行われる学習方法の 1 つであり、協調学習による豊富な学習効果は教育分野において古くから認識されている。協調学習は、同一の課題について学習者が意見交換、競合、交渉・合意形成等を繰り返し、グループの合意としての成果を出す[1]。その過程では、個々の意見の調整、修正などが行われる。ここで重要視されるのは、最終的な成果を出すに至るまでのプロセスや、他者との相互作用の質である。最終的に問題が解決されたとしても、そこに至るまでのプロセスによって学習者が獲得する知識やスキルが異なるため、より効果的なプロセスを参加者に提供することを目的として支援環境が設計される[1]。

協調学習を含む協調作業を支援する環境は、国内外の研究者によって様々な形で開発されている。近年ではコンピュータに入力したテキスト・チャットを支援に活かすシステム[2]、学習者の視線やジェスチャ等の非言語情報を支援

に活かすシステム[3]が開発されている。しかし、学習者同士が意見交換や合意形成等を行うための「対話」の状況を支援に活かすシステムは、著者が探す限り多くは開発されていない。対話は、協調学習における相互作用の重要なキーとなり、より効果的な学習プロセスを提供するためには、対話を活かした支援も必要と考える。

これを達成するため、本研究では、まず 3 者間による対面での会話を支援することを考える。3 者間による対話では、「ある特定の 1 人がずっと話し続けていて、他参加者が話をできない状態になる」、「ある特定の 2 人だけが対話していて、それ以外の人はそれをずっと聞いているだけの状態になる」ことがある。このような状況では、対話の時間が決められていた場合、自ら持っている意見や知識を外化することができない参加者がでてきてしまう。

そこで本研究では、3 者間の対話について各参加者の発話時間を測定し、参加者全員ができるだけ均等に自らの意見や知識を外化できるような、対話を支援するシステムの開発を目的とする。支援システムは Arduino を用いて開発し、そのシステムは「5 秒以上発話をしない参加者」、「20 秒以上発話をしない参加者」、「10 秒以上発話を続けている

^{†1} 東京農工大学
Tokyo University of Agriculture and Technology

参加者」を支援対象と判断する。システムに接続されている LED ランプを光らせることによって、参加者に支援対象であることを知らせ、これにより、参加者全員が均等に意見等を外化できるようにすることを目標とした。

以下、本論文の構成は次の通りである。まず第 2 章において、本研究に関連する研究を述べる。次に第 3 章において Arduino について説明し、第 4 章で Arduino を用いて開発した支援システムの概要について述べる。第 5 章では支援システムにおける会話支援部について説明し、第 6 章でそれを含めたシステムの実装について述べる。第 7 章では支援システムを用いた実験とその結果について述べ、第 8 章で本研究をまとめる。

2. 関連研究

2.1 コミュニケーション分析

Kumano ら[4]は、複数人の対話における参加者の表情と視線対象に着目し、これらの非言語情報が付与された対話データを分析する手法を提案している。Brennan[5]は遠隔地の共同作業における発話と支援情報の共有がもたらす作業効率の違いを調査し、視線共有が作業効率と関係していることを述べている。林ら[6]は、協調学習の効果的な分析に向けた非言語情報に基づく学習者の学習態度可視化システムを提案した。林らは、学習者の注視対象情報、発話区間情報、ノート記述動作情報に注目している。3 者間の協調学習を支援するためには対話に限らず、非言語情報についても考慮する必要があることが示唆されている。本研究では、まず参加者の対話状況から支援することを考える。

協調作業における対話分析として、平井ら[7]はペアプログラミング学習中における 2 者間の対話分析を行い、制限時間を与えられた課題において、発話長や連続発話が問題解決の可否と関係していることを報告している。また、Hu ら[8]もペアプログラミング学習中における 2 者間の分析を行い、問題解決が成功したペアは、発話の直前と直後にプログラム記述をしていたと報告している。本研究では、3 者間の対話支援のため、2 者間の対話分析結果をそのまま活かすことはできないものの、これらの研究から協調作業における問題解決の可否は対話がキーになる可能性を示唆している。

2.2 協調作業支援システム

3 者間による協調作業や協調学習を支援するシステムはこれまで多く開発されている。羽山ら[9]は、対面型協調学習の中で事前学習に用いた資料の適切な利用を促進させることで、議論の質を高める技術について述べている。音声認識インターフェースを利用して得られた対話テキストから対話状況を判断し、その状況に関連する事前学習で用いた資料のランキングを逐次提供している。Kim[10]らは、グループの対話状況から、現在のグループのコラボレーションの様子を可視化する Meeting Mediator を開発している。

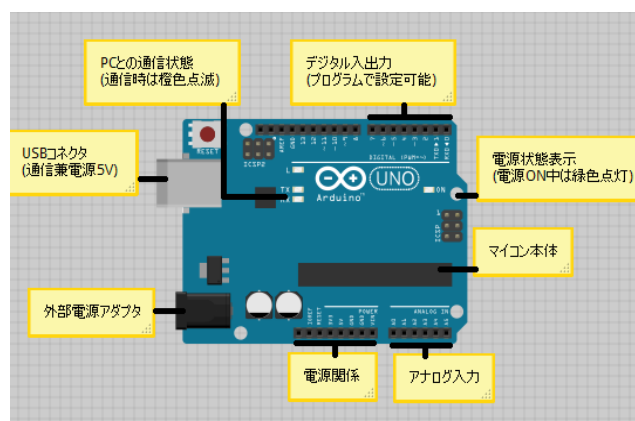


図 1 Arduino の構成要素

Figure 1 Component of an Arduino

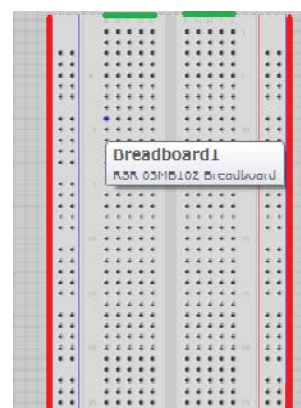


図 2 ブレッドボード内の配線

Figure 2 Wiring on a breadboard

これらのシステムは参加者の対話状況から、現在の参加者間の協調性を推定し、参加者にそれをフィードバックすることで協調学習を支援している。しかし、これらのシステムはフィードバックする内容が複雑であり、参加者間の協調性を確認するためには、ある程度システムに注視しなければならず、視線が自然とシステムの方を向いてしまう。これにより、参加者間の協調性は小さくなる可能性がある。

本研究では、対話支援システムに注視することなく、直感的に、参加者間の協調性を把握することができるシステムの開発を目指す。

3. Arduino

Arduino は、8bit のマイコンが搭載されたデジタル入出力用装置である。6 本のアナログ入力ピンと 14 本のデジタル入出力ピンが提供されており、PC と Arduino における双方向通信を利用することで、様々な制御を促すことができる。また、32KB のフラッシュメモリを搭載しているため、PC の電源を切っても、Arduino 上に実装したプログラムデータを保存しておくことができる。Arduino にある USB コネクタを、USB ケーブルで PC と接続することによって、PC 側で開発したプログラムのアップロードが可能となる。本

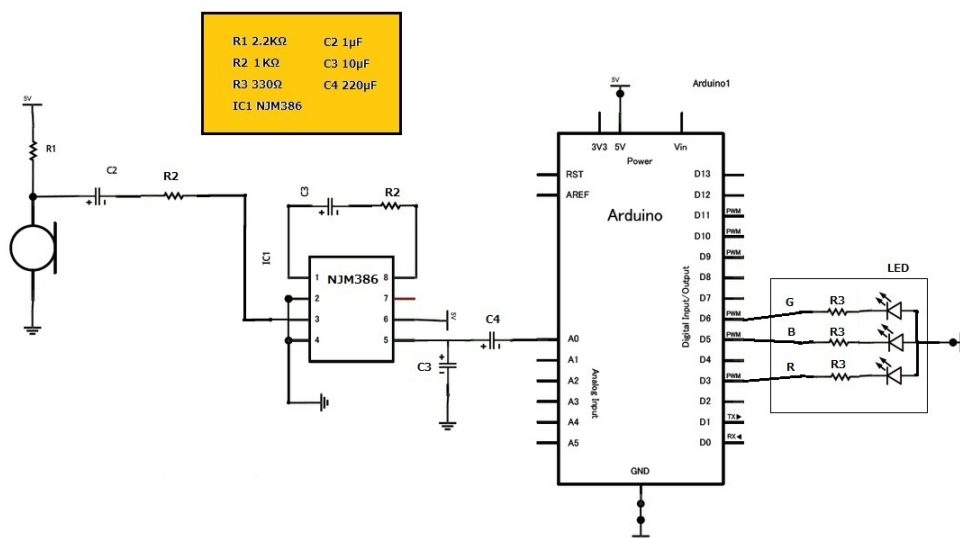


図 3 音声入力装置の配線図

Figure 3 Our circuit for voice input

研究では図 1 に示す Arduino-UNO を用いて実装した。

Arduino などのマイコンシステムの開発では、先に回路を設計してから中身のプログラムを実装し、コンパイル、実行という手順で行われることが一般的である。プログラムの実装には、総合情報開発環境「スケッチ」を用いた。スケッチで書いたプログラムが、Arduino の制御を司ることになる。実装したプログラムを PC 上でコンパイルし、USB ポートを通して、Arduino に送ることによって、そのプログラムと設計した回路に応じた挙動をする。

システムを実装するためには、Arduino だけでなく、それに付随した I/O 機器（以下、回路とする）が必要となる。回路を設計するときに必要な不可欠なのが、回路全体の基盤となるブレッドボードである。ブレッドボードには、あらかじめ配線のルールが決まっている。図 2 にある赤の実線と緑の実線が、ブレッドボードにおける中のつながりを表している。つまり、図 2 の左右から見て外側 ($2 \times n$) は縦のライン（赤の実線）でつながっており、残りの内側 ($10 \times n$) の穴が横のライン（緑の実線）でつながっていることを意味する。ここで n とは、ブレッドボード縦 1 列分の穴の最大数のことである。これを考慮して、ブレッドボードを使った回路の実装を進める。

4. Arduino を用いたシステムの開発

4.1 システム概要

本システムには、「音声入力」、「音声値認識」、「LED 出力（支援実行）」の 3 つの機能を搭載する。音声入力の部分が、人の発話音声を PC に送る。音声値認識の部分が、送られてきた値から、その人の発話状態を判断、支援対象かどうかを決定する。支援が必要であると判断した場合は、出力処理をする。LED が出力され、その支援対象となった

人に対して、現在の発話状態を改善するように促す。

4.2 音声入力

音声入力には、発話者の音声を電気信号に変換して、Arduino へ入力値として送る必要がある。そのための「コンデンサ・マイクロセンサ」に加えて、読み取る値の精度を高めるために、読み込んだ入力値を安定させるための「オーディオアンプ」を用いる。これらを、ブレッドボードに接続し、図 3 に示す配線図のように組み立てることで、音声入力機能を持つ回路となる。

マイクロセンサの一侧をグラウンドと接続し、+側を電源（ここでは 5V）と接続する。電源から供給される電流の大きさを抑えるために $2.2k\Omega$ の抵抗を間に挟んでいる。入力値の安定のために使用するアンプとコンデンサは図 3 のとおりにつなぐ。なお、ここで使用する抵抗とコンデンサ、アンプの型番は次のとおりである。

- R1... $2.2k\Omega$
- R2... $1k\Omega$
- アンプ...NJM386
- C2... $1\mu F$
- C3... $10\mu F$
- C4... $220\mu F$

これらの抵抗値とアンプ、コンデンサの値を使用するにあたって、文献[11]で挙げている回路で使われている機器を、本システム実装の参考にした。また抵抗値は、動作実験を進めていく中で、最適な環境となったものを選択した。

4.3 音声値認識

音声値認識は、PC 内の総合開発環境「スケッチ」で実装したプログラムを組み込んだ Arduino が行う。マイクロセンサからの値を Arduino が読み取るためには、次の命令を書く必要がある：

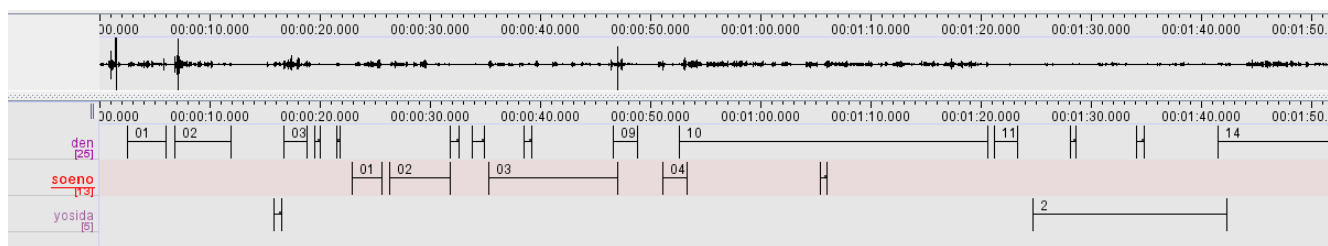


図 4 ELAN を用いた会話データの分析

Figure 4 Analysis of communication data by ELAN

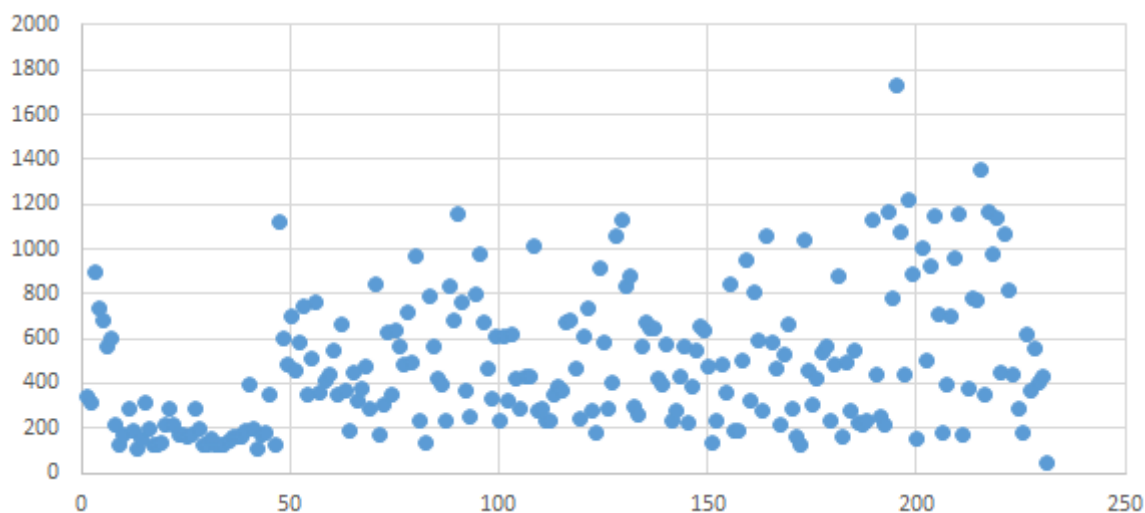


図 5 1 秒間の絶対値の合計（横軸の単位：秒）

Figure 5 Summation of absolute values of a second

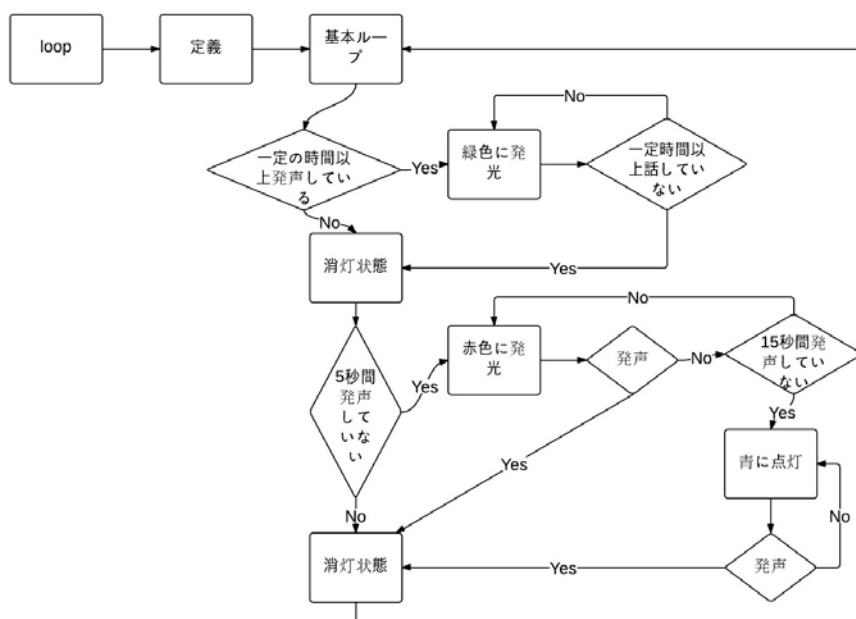


図 6 システムのフローチャート

Figure 6 The flowchart of our system

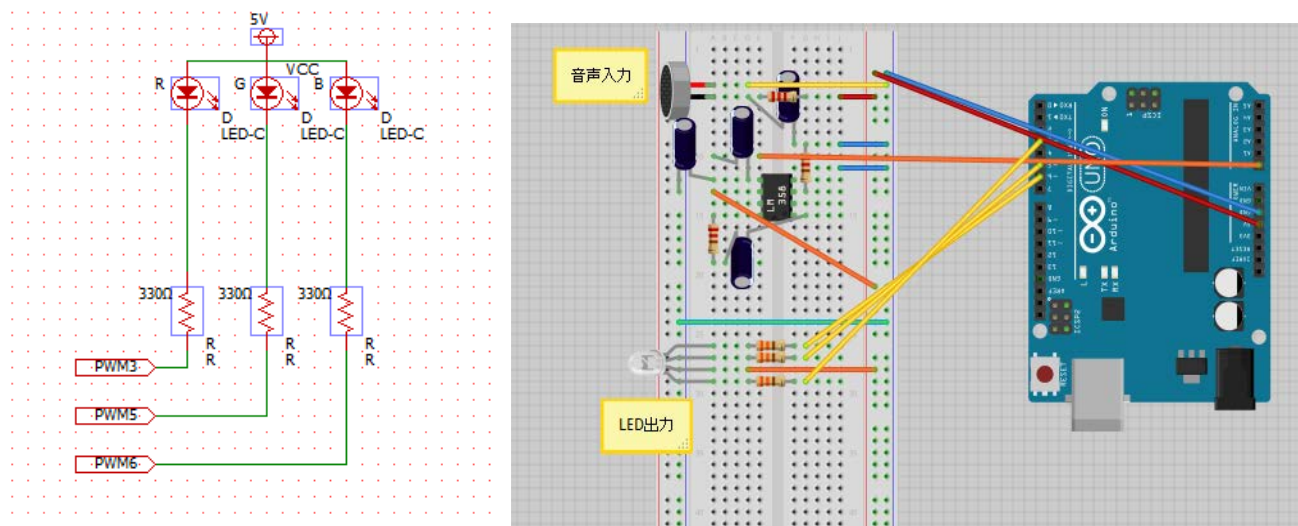


図 7 システムの実装

Figure 7 Implementation of our system

「analogRead (ピン番号) : 返り値 int 型」

この命令によって、発話音声を入力値 (int 型) として読み取る。これを、一人ひとりの発話状態を値として見ることができ、コンピュータ内での比較や演算が容易となる。この比較や演算によって、その人の現在の発話状態を認識していくことになる。それには、次の命令を書く必要がある：

「If (条件式) {出力処理 ; }」

本システムでは、この条件文を複数使用する。

5. 会話支援部

5.1 予備実験

会話支援をする対象者を決定するため、予備実験を実施した。予備実験は、大学生 9 名を対象として行った。その 9 名をランダムに 3 人ずつ 3 組に分け、決められたテーマについて 5 分間会話してもらうよう指示した。テーマは「なぞなぞを解く」というクローズドな問題の解決とした。それぞれ会話を録音し、アノテーションツール ELAN[12]を用いて、それぞれの会話データを分析した。ELAN では音声に対して注釈を付ける事ができる。ELAN を使い、それぞれの会話データにおいて、誰がどのくらい話したかの注釈を付けることで行った。分析例を図 4 に示す。

これらのデータについて分析を行った結果、実験者本人の恣意的な解釈となるが、約 10 秒以上 1 人が発話している状態は、話しすぎているのではないかと違和感を覚え、5 秒以上 3 人が喋っていない場合スムーズに会話できていないという感覚を覚えた。また、1 人が 20 秒以上発話していない時はその人物に対してかなり違和感を覚えた。これらの結果を踏まえ、本研究では、ある 1 人について、次の 4 つの発話状態が存在すると仮定した。

I. 5 秒以上発話していない

II. 20 秒以上発話していない

III. 10 秒以上発話している

IV. その他 (普通に発話している)

ここで I, II, III を満たす会話参加者を「会話支援対象」と位置付けた。

次に、「発話している」、「発話していない」状態を判断する基準を定めた。50ms ずつマイクからの音声値を取得し、その音声値を静音と判断する基準値との絶対値の差を取って加算していき、1 秒間 (50ms×20 回) 分の合計値が「発話している」とされる値を超えた場合、「発話している」と判定するプログラムを作成することにした。

これらの基準値についてもデータを分析して定めた。まず、静音とされる場合のマイクの音声値であるが、これは静かな空間に装置を放置してマイクの音声値を取得し、その後得られた値を平均して求めた。その結果、基準値として「525」が求められた。次に、発話しているとされる 1 秒間の差の絶対値の合計値である。これは喋っている状態でのマイクの音声値を取得、それを図 5 に示すように 1 秒間の合計値として出力していき、喋っている場合にその値を超えている場所を求めた。その結果「400」を求めた。システム実装の際は、それらの値を基準値として使用した。

以上より 4 つの発話状態を実装するスケッチのプログラムを作成した。そのフローチャートを図 6 に示す。フローチャート内の発光・消灯については 5.2 節で説明する。

5.2 支援の実行

システムによる認識結果を返すための出力装置には、ブレッドボードに接続した「発光ダイオード (以下、フルカラー LED)」を使用する。これは、認識結果を端的に表現するものとして、可視化できるものが最適だと考えたためである。Arduino で認識した会話参加者の発話状態を、各条

(Arduino のデジタル入力・出力部分の定義)

```
void setup() {LED のセットアップ}

void loop() {
  {変数の宣言}—①

  for() {
    /* 1 秒間経っていた場合、発話状態の条件判定・
    変数の更新 */ —②
    if() {} else {}
    /* 1 秒間経っていない場合、変数を更新する */
    if() {}

    if("10 秒以上喋っている場合") {
      for() {
        /* 緑色に点灯 */
        /* ②, ③と同様の条件判定を行い喋っていな
        かった場合ループを抜ける */
        delay(50)
      }
    }

    if("5 秒以上喋っていない場合") {
      for() {
        /* 赤色に点灯 */
        /* ②, ③と同様の条件判定を行い喋っていた
        場合ループを抜ける */
        if("15 秒以上喋っていない場合 (前条件の 5
        秒と合わせて 20 秒)") {
          for() {
            /* 青色に点灯 */
            /* ②, ③と同様の条件判定を行い喋って
            いた場合ループを抜ける */
            delay(50)
          }
        }
        delay(50)
      }
    }
    /* 支援の必要がない場合の消灯処理 */
    delay(50)
  }
  delay(50)
}
```

図 8 プログラムの概略
 Figure 8 Outline of our program

件文と比較して、支援対象となる規定秒数に到達した場合、その秒数に達した会話参加者の回路に出力命令を送る。

LED には極性があるので、足が長い「アノード」という部分から、足が短い「カソード」という部分への 1 方向の電流しか流すことができない。つまり、電圧が高いほうをアノードに接続する必要がある。また、LED に電流が流れすぎないように、流れる電流に制限をかける必要がある。そこで、330Ω の抵抗を 3 つ用意し、3 つのカソードと出力ピンの間に挟ませるようにする。フルカラー LED を出力するための配線の詳細は、図 7 左のとおりである。フルカラー LED の RGB 線にはそれぞれ、330Ω の抵抗を挟み、デジタル入力ピンに接続する。足の長いアノード側は電源（ここでは 5V）ピンに接続し、電源を供給するようにする。

表 1 主な変数・関数の説明

Table 1 Explanation of variables and functions used in our program

名称	用途
delay()	() の中に書かれた[数字]ms だけ処理を遅延する関数
Loop1	5 秒以上喋っていない場合の条件分岐に使用する変数。本プログラムでは喋っていないと判定された場合 1 秒で 1 加算され、喋っていると判断された場合 0 にリセットされる。
Loop2	20 秒以上喋っていない場合の条件分岐に使用される変数。
level	音声センサの値を保持する変数。ループが始まるごとに呼び出され値が更新される。
SEC	1 秒間の値を保持する変数。
heni	1 ループごとに、音声の数値と基準値との差を加算していく変数。
zettai	heni に加算する際の、音声の数値と基準値との絶対値を保持する変数。
analogWrite (‘LED 名’, ‘数値’)	‘LED 名’の LED の値を‘数値’にする関数。これを RGB それぞれについて記述することで色の点灯を実現している。

```
level=analogRead(SOUND_SENSOR0);
/*SEC が 20 以上=1 秒間分の変位を取得したら*/
if(SEC >= 20){
  if(heni > 400){ /* 喋ってる */
    Loop3++;
    Loop1 = 0;
    SEC = 0;
    heni = 0;
  }else{ /* 喋ってない */
    Loop1++;
    Loop3 = 0;
    SEC = 0;
    heni = 0;
  }
}
/*SEC が 20 以下=1 秒経ってない*/
if(SEC < 20){
  zettai = abs(level - 525);
  heni = heni + zettai;
  SEC++;
}
```

図 9 1 秒間の変位を取得してパラメータの更新を行うプログラム

Figure 9 The program for updating parameters when getting displacement in a second

6. 実装

6.1 回路実装

音声入力と LED 出力の両方の機能を持つ回路を作るため、ブレッドボードに機器を図 7 右のように組み込んだ。

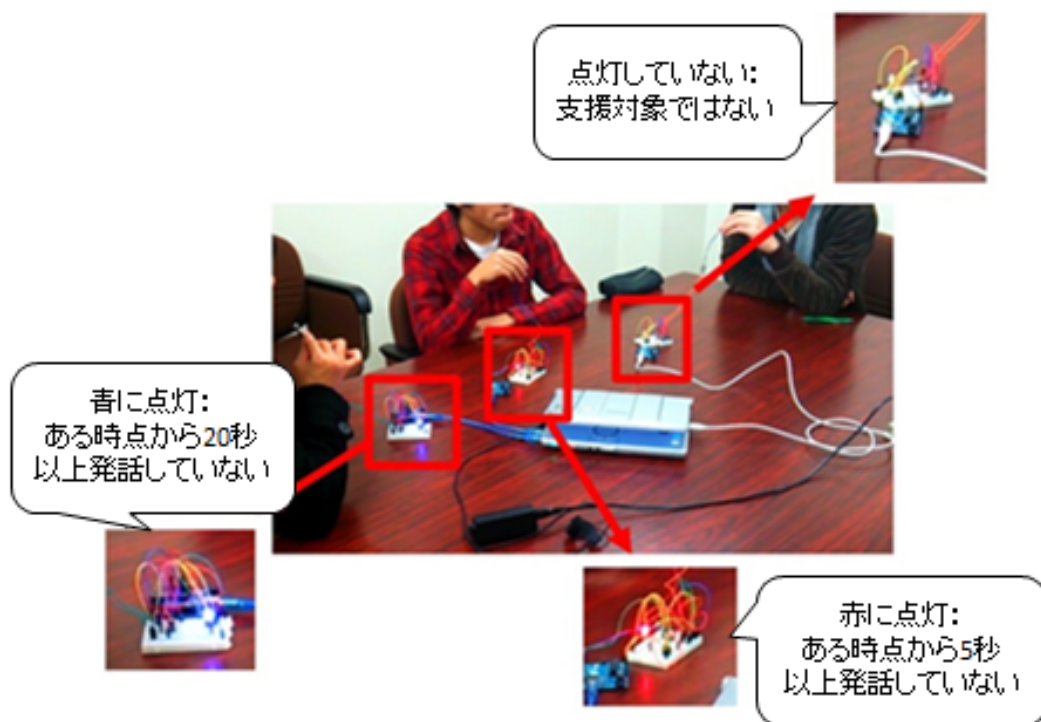


図 10 システム利用の様子

Figure 10 A screenshot of conversation using our system

だ. このように回路を組むことによって, 音声入力, 出力処理のどちらにも対応した 1 つの回路ができあがる.

6.2 スケッチ実装

まず, プログラムの全体を把握するための概略を図 8 に示す. 図 8 内の①で示している変数の宣言の際に宣言している変数の説明と, 全てのループに組み込まれていて, 必ず一周ごとに一回実行される関数である `delay(50)`, LED の点灯を操作する関数である `analogWrite` についての説明を表 1 に示す.

図 8 での②に当たる 1 秒間の変位を取得する部分について説明する. まず図 9 に該当部分の詳細なプログラムを示す. ここでは, まず `level` に `analogRead` で取り出したマイクの値を代入している. 次に 1 秒間を判定する変数 `SEC` が 20 以上 (一周 50ms のループなので $50 \times 20 = 1000$ より 1 秒) の場合と 20 未満の場合の場合分けを行っている. `SEC` が 20 未満の場合は, 1 秒間経過していないので, `zettai` に `level` と基準値 (本プログラムの場合 525) の差の絶対値を代入, そしてそれを `heni` に加算し, `SEC` を 1 増やしている. `SEC` が 20 以上 (この場合, 想定しているのは 20 になった瞬間 (1 秒ちょうど) であるが, バグを回避するために 20 以上とした) となった場合は, `heni` の値が喋っていると判定する閾値 (本プログラムの場合 400) を超えているかいないか判定し, 超えていた場合は喋っていると, `Loop3` に 1

を加算, `Loop1` をリセットして喋っている状況に推移させ, `heni`, `SEC` をリセットしている. 閾値を超えていなかった場合は喋っていないとし, `Loop1` に 1 を加算, `Loop3` をリセットし, 喋っていない状況に推移させている.

7. 利用実験

7.1 実験概要と結果

本研究で開発したシステムが正しく動作するかどうかを確認するため, 大学生 3 名を実験参加者としたシステム利用実験を行った. システムを利用した会話の様子を図 10 に示す. 各実験参加者に 1 台のシステムを割り当て, それに接続されているマイクを持って自由に会話するよう指示した. 各実験参加者用のシステムに接続されている LED の発光色によって, その実験参加者の発話状況を把握することができる.

各実験参加者に対し, システム利用の感想を聞いた. 全員「役にたった」という回答が得られ, 特に「誰が喋っているか, いないか」を知ることが出来るのが良い点であるという意見が目立った. しかし, 点灯条件や点灯した後の会話の促しを感じられないことについて改善して欲しいという意見が見られた. 以下, 各実験参加者の感想の内容を示す.

- 実験参加者 A (男性)

誰が喋っているか、全然喋っていないのかが分かるため、システムは会話の役に立った。ただ、誰が喋っているかが分かっているにもかかわらず具体的などうしたらいいのかが分からないところは改善すべき点だと思った。(A だけがしゃべっていて、B と C が全くしゃべっていなかったとき、どのタイミングで B (あるいは C) がしゃべり始めればよいのか)

● 実験参加者 B (男性)

システムのおかげで自身の発話時間を気にするようになるので、喋りすぎまたは喋らなさすぎを注意するようになった。LED ライトの点灯条件を再検討する必要があると思う。特に喋っていない時の改善が必要。会話をする際に自分の考えをまとめるには 5 秒という時間が短すぎると感じたときもあった。そのためもう少し点灯条件を緩めに設定したほうがよい。

● 実験参加者 C (女性)

話せなかったけど、話さないといけない気持ちになった。ピッチ変更ができるといいと思う。

7.2 改善点

今回使用したマイクロホンでは、発話していないときに、発話していると誤認識する場合があった。これは、実験の会話とは別の音 (ノイズ) を拾ってしまったためと考えられる。ノイズの駆除には、より高性能なマイクロホンを使う、またはアンプの数を増やすといった対策をすることで、より精度の高い音声入力装置となる。

開発したシステムでは、低音の人の発声はよく拾うのに対して、高音の人の発声を拾いづらく、発話しているにも関わらず、発話していないと認識する場合がやや多く見られた。これは、スケッチで実装したプログラムの設定した閾値による問題が考えられる。本開発では、恣意的な判断で閾値設定を行った要素が強かったため、この閾値を確かなデータに基づいて設計することで、人に左右されない確かなシステム開発につなげることができる。

開発したシステムでは、認識結果を可視化できるものが適切と考え、フルカラー LED を用いたが、視覚障害のある方には適切な出力処理とは言えない。これを音声による出力処理で支援するという手もあるが、それは聴覚に障害のある方に不適切である。あるいは、バラエティ番組などで用いられる電流を微量流して発話者に痛覚として感知させることもできる。これに関しては、多種多様なユーザが利用できるという観点を目指す上でも、さらなる熟考が必要である。

このシステムを有効利用するにはどのような場面が挙げられるかを考えた。「知識の外化の均等化」が本研究の目的なので、例えば、外国人との会話学習などの場面で使うのが良いのではないかと考える。英会話学習において、英語を教える 1 人と教わる 2 人での場面を想定する。このとき、2 人が英会話できずに詰まっている状態を、このシステムで知らせ、英会話の知識外化を促す。これによって、

英会話学習の向上に役立つ可能性がある。

8. おわりに

本研究では、3 者間の対話において、参加者の発話時間から、対話を支援するシステムを、Arduino を用いて開発した。本支援システムを用いることにより、支援システムを注視することなく、参加者自身や他の参加者の発話状況を直感的に把握することを目指した。利用実験の結果から、実験参加者から「誰が発話しているか否かを知ることができる」という意見が得られ、すべての実験参加者からシステムが対話の役に立ったという意見が得られた。一方で点灯条件や点灯したあとの会話の促しが感じられない等の改善点に関する意見も得られた。

今後は、対話支援方法についてさらに検討を行ってシステムを改善し、本支援システムを有効に活用できる場面について検討を行う予定である。

謝辞 本研究に協力いただいた東京農工大学向井翼氏に感謝いたします。本研究の一部は科学研究費補助金若手研究 (B) 25870207 による。

参考文献

- 1) 稲葉晶子, 豊田順一: CSCL の背景と研究動向, 教育システム情報学会誌, Vol. 16, No. 3, pp. 111-120 (1999).
- 2) 林佑樹, 小尻智子, 渡辺豊英: 「貢献への気づきを反映した議論支援インターフェース」, 情報処理学会論文誌, Vol. 53, No. 4, pp. 1461-1471 (2012).
- 3) Nawahdah, M. and Inoue, T.: Motion adaptive orientation adjustment of a virtual teacher to support physical task learning, Journal of Information Processing, Vol. 20, No. 1, pp. 277-286 (2012).
- 4) Kumano, S., Otsuka, K., Mikami, D., Matsuda, M., and Yamano, J.: Understanding communicative Emotions from
- 5) Brennan, S. E., Chen, X., Dickinson, C. A., Neider, M. B., and Zelinsky, G. J.: Coordinating cognition: the costs and benefits of shared gaze during collaborative search, Cognition, Vol. 106, No. 3, pp. 1465-1477 (2008).
- 6) 林佑樹, 小川裕史, 中野有紀子: 協調学習における非言語情報に基づく学習態度の可視化, 情報処理学会論文誌, Vol. 55, No. 1, pp. 189-198 (2014).
- 7) 平井佑樹, 井上智雄: ペアプログラミング学習における状態の推定—つまずき解決の成功と失敗に見られる会話の違い—, 情報処理学会論文誌, Vol. 53, No. 1, pp. 72-80 (2012).
- 8) Hu, J., Inoue, T.: Investigating Pair Programming Learning for Seeking Success Factors in Collaboration, IPSJ SIG Technical Reports Vol. 2013-GN-87, No. 14, pp. 1-8 (2013).
- 9) 羽山徹彩, 徐利娟, 國藤進: 議論での事前学習知識の活用を促す対面型協調学習支援システム, Vol. 55, No. 1, pp. 163-176 (2014).
- 10) Kim, T., Chang, A., Holland, L., Pentland, A. S.: Meeting Mediator: Enhancing Group Collaboration using Sociometric Feedback, Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work, pp. 457-466 (2008).
- 11) NJM386 データシート: (Online Available: 2014/02/16) http://semicon.njr.co.jp/jpn/PDF/NJM386_J.pdf.
- 12) アノテーション支援ツール ELAN: (Online Available: 2014/02/16) <http://www.lat-mpi.eu/tools/elan>