

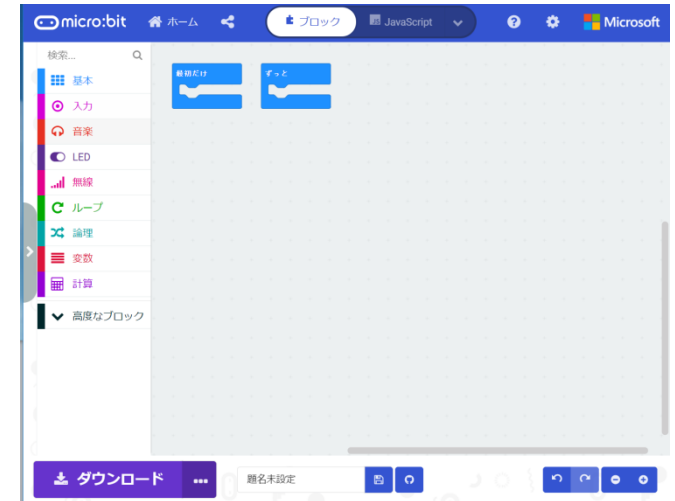
IP_NETWORK 仕様書

micro:bitを用いてIPアドレスを定めてネットワークを用いた
双方向的なコンテンツの授業の実現のための拡張

使用方法

Google等の検索エンジンを用いて
「makecode」を検索→micro:bitをクリック

高度なブロック→拡張機能→
「https://github.com/sakai-code/ip_network」
と打ち込む



もしくは「https://github.com/sakai-code/micro-bit_extention_package_by-sakai-code」からzipファイルをDL&解凍し中にある(microbit-ip_network.hex)ファイルを、高度なブロック→拡張機能→「ファイルを読み込む」から導入してください。

すると



このようなブロックが出現します。

別途必要(推奨)な物

- Phコネクタを用いた乾電池で動作をさせると持ち運びしやすくなり便利です。
- Micro:bitの標準搭載のLEDを用いて文字列を表示すると動作が遅い&他の動作を受け付けないという癖があるため、SSD1306というi2cのOLEDを用いた表示を推奨しています。その他16x2の液晶もありますがmicro:bitの動作との相性が悪いため非推奨としています。

- 使用しているSSD1306用の拡張:

URL「<https://github.com/sakai-code/oled-ssd1306>」

同じように高度なブロック→拡張機能のURL欄に貼り付けて使用します。

詳しい使い方は(仮)より

もしくは:URL

https://github.com/sakai-code/micro-bit_extention_package_by-sakai-code

内のmicrobit-OLED-ssd1306.hex を高度なブロック→拡張機能→ファイルを読み込むより使用してください。



基本仕様

• LAN

LAN

グループ番号 でデバイスのIPアドレスを192.168.0. にする

自分のIPアドレスの192.168.0.○に設定した数字を表示

デバイス宛に文字列 を受け取ったら実行する

受信した直前の相手のIPアドレス宛に文字列 を返す。

192.168.0. に文字列 (17文字までの英数字)を送信

受信した文字列 (英数字)

受信した相手のIPアドレスの情報 (192.168.0.X形式)

サーバーに ID のデータを問い合わせる

192.168.0. に ID のデータを問い合わせる

▪ SERVER

SERVER

グループ番号 、192.168.0.0のサーバーになる

グループ内で情報のやり取りがあったら"

IPアドレス+メッセージの内容の文字列をシリアル通信で出力

送出元IPアドレス to 宛先IPアドレス+文字列

▪ LAN&SERVER

LAN&SERVER

データをセットしておく

ID 文字列 (15文字までの英数字)をセット

ID に数字 をセット (自動的に文字列として保存)

この三つの機能に分かれており、LANが主に使用する機能で、SERVERが先生やグループ内1名がグループ内のメッセージのやり取りを見る際に使用する機能です。

LAN&SERVERはデータ(文字列や温度などの情報)を登録しリクエストがあった場合に登録してあるIDの情報を返す機能です。SERVERはなくても動作します。

LAN機能解説(1)

グループ番号 **1** でデバイスのIPアドレスを192.168.0. **1** にする

- グループ番号とIPアドレス決めます。最初だけブロック内に配置します。グループ番号が違うとやりとりをすることが出来ません。同じIPアドレスのデバイスが存在すると正常に動作しません。

自分のIPアドレスの192.168.0.○に設定した数字を表示

- 先ほど定めた192.168.0.○の○に入力した数字をmicro:bitのLED上に表示します。

デバイス宛に文字列 **receivedtext** を受け取ったら実行する

- 定めたグループ番号とIPアドレス(192.168.0.○)宛に文字列(例:“HELLO”)などを受信した際にこのブロック内のプログラムが実行されます。赤い**receivedtext**は受信した文字列です。受信した文字列は後程解説するLAN&SERVER機能の相手のデータをリクエストした際の返信された文字列(例“ABC”)このreceivedtext内に格納されます。

LAN機能解説(2)

受信した直前の相手のIPアドレス宛に文字列 " " を返す。

- 最後に文字列を送信してきた相手のデバイス宛(例192.168.0.2)に文字列を送信します。
デバイス宛に文字列 receivedtext を受け取ったら実行する 内に置くことで自動的に返信することができます。

192.168.0. 1 に文字列 "Hello!" (17文字までの英数字)を送信

- おなじグループ内の192.168.0.○の○に入力したデバイス宛に文字列(例:"HELLO")を送信します。自分のデバイス宛に送信した場合何も起きません。

受信した文字列 (英数字)

- 最後に受信した文字列です(例:"HELLO!")

受信した相手のIPアドレスの情報 (192.168.0.X形式)

- 最後に受信した相手のIPアドレスの情報です。(例:"192.168.0.2") OLED上などにどの相手がメッセージを送ってきたかをわかりやすくする際に使用します。

LAN機能解説(3)

サーバーに ID 0 のデータを問い合わせる

- LAN & SERVER機能内でサーバー(192.168.0.0)に選択したID番号のデータをリクエストします。サーバーが存在しない場合何も返ってきません。また、データが登録されていない場合:“NODATA”が返ってきます。

192.168.0. 1 に ID 0 のデータを問い合わせる

- LAN & SERVER機能内で相手のデバイス(192.168.0.〇)に登録されている選択したID番号のデータをリクエストします。相手が存在しない場合何も返ってきません。また、データが登録されていない場合:“NODATA”が返ってきます。
- なお上記で問い合わせたデータは全て

デバイス宛に文字列 receivedtext を受け取ったら実行する

ブロックのreceivedtextへ格納されます。

SERVER機能解説(1)

グループ番号 1、192.168.0.0のサーバーになる

- 最初だけブロック内にいれると、入力したグループ番号内でIPアドレス:192.168.0.0のサーバーとなりメッセージの流れをモニタリングすることが出来ます。また後程解説するLAN&SERVER機能内でデータを登録しておくと、リクエストがあった時にデータを返すことが出来ます。

グループ内で情報のやり取りがあったら"

- グループ内で情報(文字列)のやり取りがあったらこのブロック内のプログラムが実行されます。

IPアドレス+メッセージの内容の文字列をシリアル通信で出力

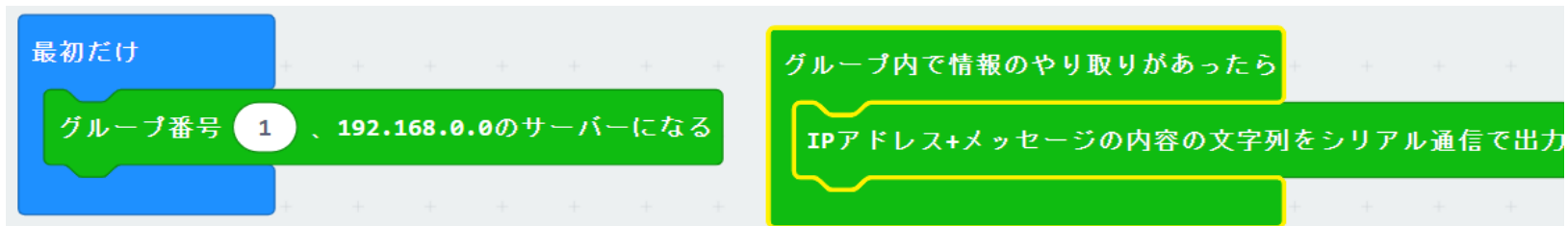
- やりとりの内容をシリアル通信出力で表示します。シリアル通信の使い方は次のページで解説しています。

送出元IPアドレス to 宛先IPアドレス+文字列

- 送信元のIPアドレス to 宛先IPアドレス + 文字列形式の文字列を出力します。

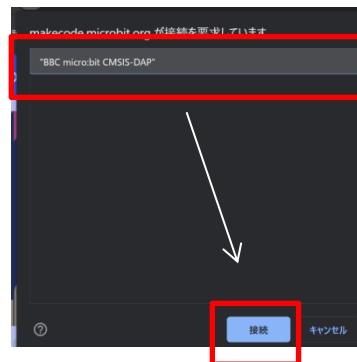
SERVER機能解説(2)

- SERVER機能内のメッセージのやりとりをシリアルモニタでモニタリングする方法を解説します。ブロックをこのように配置します。



- 例: グループ1のサーバー(192.168.0.0)となりメッセージの流れを見ている

- micro:bitをUSBでPCに接続→
- ダウンロードの右...ボタンをクリックしデバイスを接続するをクリック→



SEVER機能解説(3)

- シリアル通信のモニタリング機能の解説の続き

- メッセージのやり取りがグループ内であると→
が表示される。クリックすると下の画面が表示されます。

||| コンソールを表示 デバイス



```
===192.168.0.2===|===192.168.0.1===|===REQUESTDATA:4===|  
|=====|=====|=====|  
RECEIVED!  
|....[FROM].....|....[TO].....|....[MESSAGE]....|  
|===192.168.0.2===|===192.168.0.1===|===REQUESTDATA:4===|  
|=====|=====|=====|  
RECEIVED!  
|....[FROM].....|....[TO].....|....[MESSAGE]....|  
|===192.168.0.2===|===192.168.0.1===|===REQUESTDATA:1===|  
|=====|=====|=====|
```

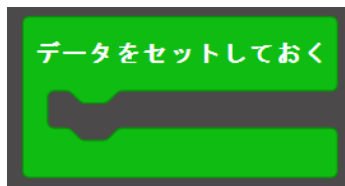
送信元アドレス | 宛先アドレス | 文字列の内容

上記の形式となっています。文字列の内容がREQUESTDATAの場合 登録されているIDの情報を問い合わせている状態です。

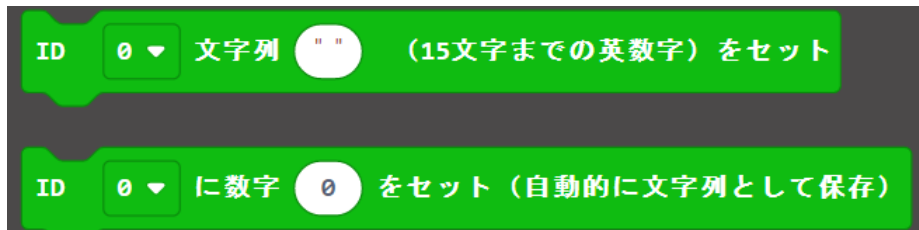
```
=====|=====|=====|  
RECEIVED!  
|....[FROM].....|....[TO].....|....[MESSAGE]....|  
|===192.168.0.2===|===192.168.0.1===|===REQUESTDATA:1===|  
|=====|=====|=====|  
RECEIVED!  
|....[FROM].....|....[TO].....|....[MESSAGE]....|  
|===192.168.0.2===|===192.168.0.1===|===REQUESTDATA:1===|  
|=====|=====|=====|
```

LAN&SERVER機能解説

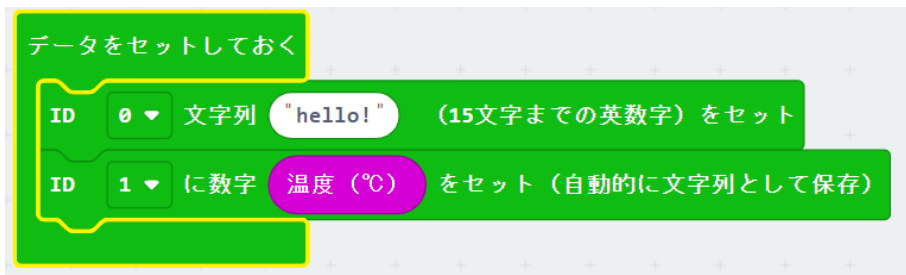
- SERVER&LAN共通の機能です。選択IDにデータを登録しておく、リクエストがあった際にその値を返します。(あらかじめサーバーかLAN機能が設定済み前提)



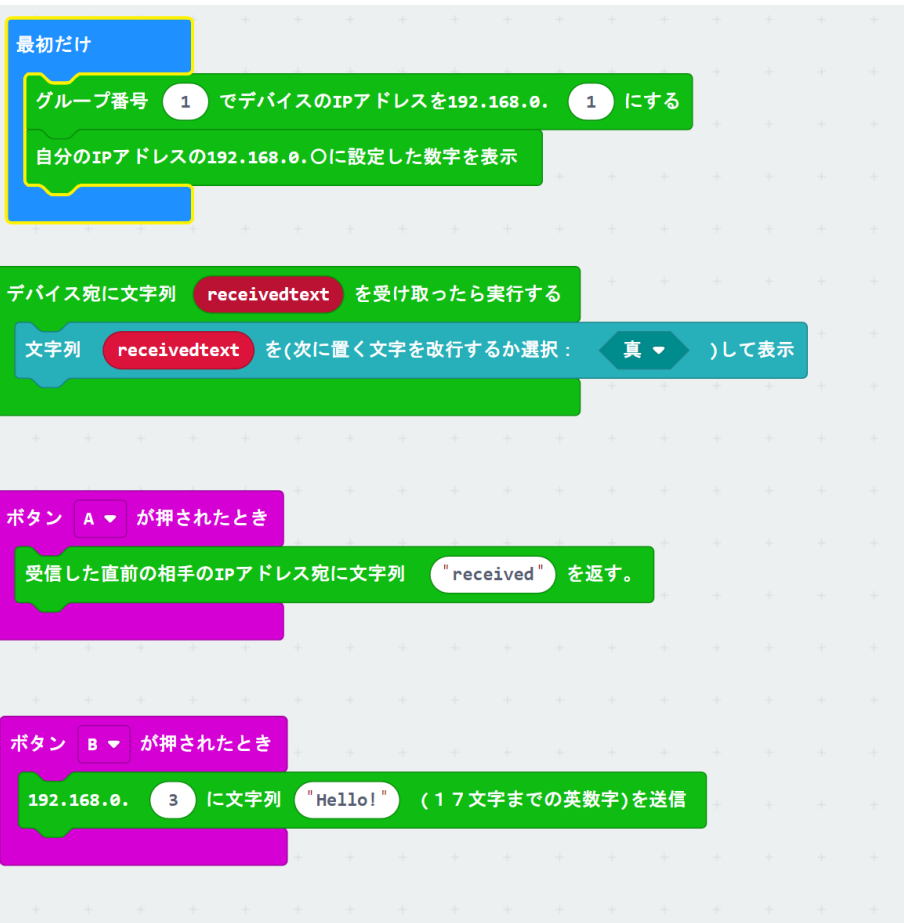
- このブロック内登録するデータを設定します。



- 選択したIDにデータをセットしておきます。リクエストがあった場合この中身を返します。 使用例 (ID:0に“hello!” ID:1に端末周辺の温度を登録している状態)



サンプルプログラム 1



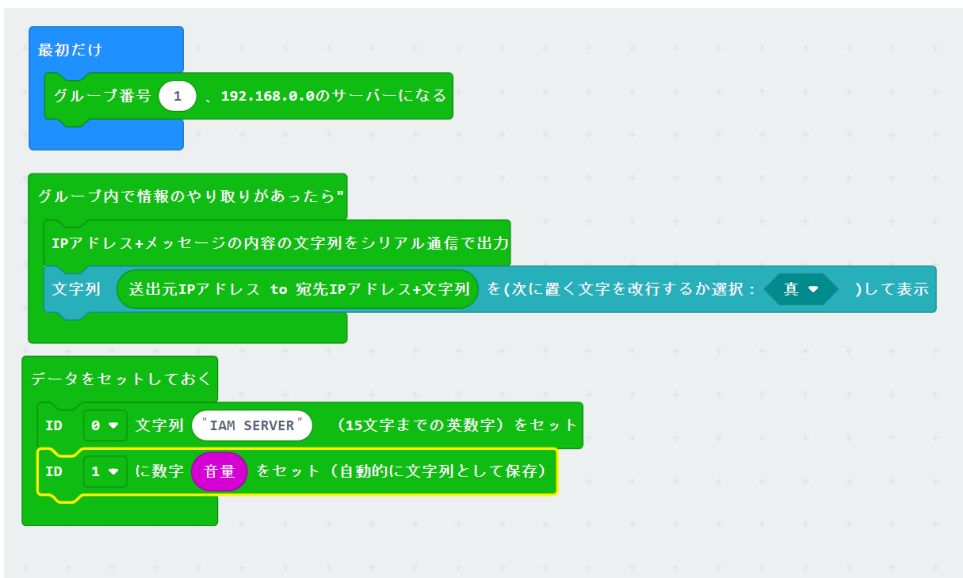
- グループ1 192.168.0.1となり1をmicro:bit上のLEDに表示
- 文字列を受信したらOLED(SSD1306)に表示
- Aボタンが押されたら直前に受信したデバイス宛に“received”を送信
- 192.168.0.3に“Hello!”を送信

サンプルプログラム2



- グループ1で192.168.0.2にする
- 文字列を受信したらOLED (SSD1306)に表示する
- ID0に“I am satou”
- ID1に周辺の明るさ
- ID2に周りの温度
- ID3にピッチ(縦方向の傾き)を登録
- ボタンAが押されたらサーバーにID:0のデータを問い合わせる
- ボタンBが押されたら192.168.0.1のデータを問い合わせる

サンプルプログラム3



- グループ1 192.168.0.0のサーバーになる
- グループ内でメッセージのやり取りがあったら シリアルモニタで内容を見る
- OLED上(ssd1306)で見る
- データをセットする
 - ID0にIAM SERVER
 - ID1に周りの音量

改良のヒント



- 受信したら音を出す



- `movemotor` が来たら回転サーボモータを50%で動作する



- etc