

Sorties

- Entrées
- Délais
- Boucles
- Variables
- Maths
- Procédures
- Tâches
- Moteurs
- Liaison série
- Avancé

sortie A.0 activée

début

sortie A.0 activée pendant 1000 ms

Faire clignoter la sortie A.0 pendant 500 ms

jouer son 64 pendant 500 sur A.0

jouer Happy Birthday sur A.0

SPE035 play tune 1 sur A.0

mélodie ;Clic droit pour démarrer l'assistant

envoyer code infrarouge 1 à A.0

positionner servo B.0 à 150

positionner servopos B.0 à 150

signal pwm de periode 100 rapport cyclique 200 sur B.0

initialiser les broches B à 255

initialiser le type des broches B à 255

basculer A.0

Sorties

Entrées

- Délais
- Boucles
- Variables
- Maths
- Procédures
- Tâches
- Moteurs
- Liaison série
- Avancé

si l'entrée A.0 est activée

faire début

si l'entrée A.0 est activée

faire

sinon

lire valeur analogique en A.0 et stocker dans varA

lire température en A.0 et stocker dans varA

lire distance ultrason en A.0 et stocker dans varA

lire distance ultrason en (mode 2 broches)
trigger A.0 echo A.0 et stocker dans varA

lire valeur infrarouge A.0 et stocker dans varA
 télécommande infrarouge (+1)

entrée A.0

pins B

Sorties
Entrées
Délais
Boucles
Variables
Maths
Procédures
Tâches
Moteurs
Liaison série
Avancé

attendre pendant 500 ms

début

mettre en veille pendant 10 s

attendre jusqu'à ce que A.0 est activée

Sorties
Entrées
Délais
Boucles
Variables
Maths
Procédures
Tâches
Moteurs
Liaison série
Avancé

répéter indéfiniment

faire

répéter

jusqu'à ce que l'entrée A.0 est activée

répéter

jusqu'à varA > 10

tant que l'entrée A.0 est activée

faire

tant que varA > 10

faire

compter avec varA de 0 jusqu'à 4 par pas de 1

faire

- Sorties
- Entrées
- Délais
- Boucles
- Variables**
- Maths
- Procédures
- Tâches
- Moteurs
- Liaison série
- Avancé

si **varA** > 10
faire début

si **varA** > 10
faire
sinon

fixer **varA** à 10

fixer **varA** valeur aléatoire

incrémenter **varA** de 1

décrémenter **varA** de 1

convertir en ASCII **varA**
100 **varB**
10 **varC**
1 **varD**

varA

varA , **varB**

entrée A.0

- Sorties
- Entrées
- Délais
- Boucles
- Variables
- Maths**
- Procédures
- Tâches
- Moteurs
- Liaison série
- Avancé

fixer **varA** à 10

début

fixer **varA** à +

1

varA

entrée A.0

+

- Sorties
- Entrées
- Délais
- Boucles
- Variables
- Maths
- Procédures**
- Tâches
- Moteurs
- Liaison série
- Avancé

sous-fonction **Nom Sous-fonction**
début

- Sorties
- Entrées
- Délais
- Boucles
- Variables
- Maths
- Procédures
- Tâches**
- Moteurs
- Liaison série
- Avancé

début

début

arrêter la tâche

suspendre la tâche 0

reprendre la tâche 0

recommencer la tâche 0

recommencer toutes les tâches

- Sorties
- Entrées
- Délais
- Boucles
- Variables
- Maths
- Procédures
- Tâches
- Moteurs**
- Liaison série
- Avancé

moteur A à avancer

début

moteur A à avancer pendant 1000 ms

BOT120 Microbot

configurer la vitesse aucun changement

CHI035

AXE023

servo moteur

servo A B.0

stop A 150

décalage A 30

servo B B.1

stop B 150

décalage B 30

AXE120

vitesse gauche

vitesse droite

K-MR-01

vitesse gauche

vitesse droite

- Sorties
- Entrées
- Délais
- Boucles
- Variables
- Maths
- Procédures
- Tâches
- Moteurs
- Liaison série**
- Avancé

debug

début

sertxd « message »

Envoyer en ASCII

sortie série à N4800 sur A.0 « message »

Envoyer en ASCII

afficher sur le LCD A.0 « message »

envoyer sur le LCD effacer en A.0

entrée série à N4800 sur A.0 en varA

Recevoir en ASCII

serrxd vers varA

Recevoir en ASCII

hsersetup B9600_8

Inverser la polarité

hserout « message »

hserin varA

« message »

- Sorties
- Entrées
- Délais
- Boucles
- Variables
- Maths
- Procédures
- Tâches
- Moteurs
- Liaison série
- Avancé

Commentaire

début

BASIC ;Utilisez Ctrl+ V pour coller le code BASIC

lire varA depuis l'EEPROM 0

écrire varA dans l'EEPROM 0

initialiser l'adresse SCL de l'i2c à Nunchuck A

initialiser l'adresse SCL de l'i2c à %10100000
 i2cfast i2cword

lire varA de i2c 0

écrire varA en i2c 0

réinitialiser tout

autoriser l'interruption quand C.0 est activée

désactiver l'interruption

déconnecter

reconnecter

exécuter le programme de la mémoire numero 0

charger ce programme dans la mémoire numero 0