In[1]:=
```
ser = DeviceOpen["Serial", {"/dev/ttyUSB0", "BaudRate" → 115200}]
close := DeviceClose[ser];
bclr := DeviceExecute[ser, "ReadFlushj"];
rbuf := DeviceReadBuffer[ser];
```

... SerialLink`SerialPortOpen: Could not open the port /dev/ttyUSB0.

Out[1]=
```
$Failed
```

In[5]:=
```
Spd$ := {"S1", "S2", "S3", "S4", "Sx"}

Cmd$ :=
 {"F", "B", "RR", "RL", "TR", "TL", "BTR", "BTL", "VR", "VL", "S", "H", "R", "RU", "RD"}

Dmd$ := {"M1", "M2", "GODT", "GSTEP", ">pos", "GF", "GB", "GRR", "GRL", "GTR", "GTL"}

Man$ := {"MAN1", "MAN2", "MAN3", "MAN4", "MAN5", "MAN6"}
```

In[9]:=
```
START := DeviceWrite[ser, {83, 84, 65, 82, 84, 13}]
FRWD := DeviceWrite[ser, {70, 82, 87, 68, 13}]
BKWD := DeviceWrite[ser, {66, 75, 87, 68, 13}]
RRWD := DeviceWrite[ser, {82, 82, 87, 68, 13}]
RLWD := DeviceWrite[ser, {82, 76, 87, 68, 13}]
GO := DeviceWrite[ser, {71, 79, 13}]
zenc := DeviceWrite[ser, {122, 101, 110, 99, 13}]
```

In[16]:=
```
F := {sCmd[1]; msg$ := " Moving Forward"}
B := { sCmd[2]; msg$ := " Moving Backward"}
RR := { sCmd[3]; msg$ = " Rotating Right"}
RL := { sCmd[4]; msg$ = " Rotating Left"}
TR := {sCmd[5]; msg$ = " Turning Right"}
TL := {sCmd[6]; msg$ = " Turning Left"}
BTR := {sCmd[7]; msg$ = " Backing to the Right"}
BTL := {sCmd[8]; msg$ = " Backing to the Left"}
VR := {sCmd[9]; msg$ = " Veering Right"}
VL := {sCmd[10]; msg$ = " Veering Left"}
S := {sCmd[11]; msg$ = " Stopped"}
H := {sCmd[12]; msg$ = " Halted"}
R := sCmd[13];
RU := {sCmd[14]; msg$ = "Ramping Up Speed"}
RD := {sCmd[15]; msg$ = "Ramping Down Speed"}
```

In[31]:=
```
S1 := {ldty = 150; rdty = 150; sSpd[1]}
S2 := {ldty = 250; rdty = 250; sSpd[2]}
S3 := {ldty = 500; rdty = 500; sSpd[3]}
S4 := {ldty = 750; rdty = 750; sSpd[4]}
```

In[35]:=
```
msg1 := " Press START to Activate Robot Control"
msg2 := " STARTED – Robot Control Active"
msg3 := " REBOOTED – Press START to Continue"
msg4 := " Exercising Principal Motions "
msg5 := " Simple Avoidance Maneuver "
msg6 := " GamePad Mode "

msg$ := msg1

te := Dynamic [Clock[{0, 600, 1}]];
```

In[43]:=
```
sSpd[no_] :=  DeviceWrite[ser, Append[ToCharacterCode[Spd$[[no]]], 13]];

sCmd[no_] := DeviceWrite[ser, Append[ToCharacterCode[Cmd$[[no]]], 13]];

sMan[no_] := DeviceWrite[ser, Append[ToCharacterCode[Man$[[no]]], 13]];

Sx[n1_, n2_] := DeviceWrite[ser, Join[{32}, ToCharacterCode[ToString[n1]], {32},
    ToCharacterCode[ToString[n2]], {32}, ToCharacterCode[Spd$[[5]]], {13}]];

GODT[n1_] :=
  DeviceWrite[ser,
   Join[ToCharacterCode[ToString[n1]], {32}, ToCharacterCode[Dmd$[[3]]], {13}]];

sDmd[no_, n1_] :=
  DeviceWrite[ser,
   Join[ToCharacterCode[ToString[n1]], {32}, ToCharacterCode[Dmd$[[no]]], {13}]];

STEP[n1_, n2_] :=
  DeviceWrite[ser, Join[{32}, ToCharacterCode[ToString[n1]], {32},
     ToCharacterCode[ToString[n2]], {32}, ToCharacterCode[Dmd$[[4]]], {13}]];



Rd$ := { ".dist", ".lenc", ".renc"};

sRd[no_] := Block[ {xe}, cbuf;
   DeviceWrite[ser, Append[ToCharacterCode[Rd$[[no]]], 13]];
   Pause[0.05]; xe := DeviceReadBuffer[ser];
    FromDigits[ FromCharacterCode[Select[xe, # > 47 && # < 58 &]] ]] ;
```

In[52]:=
```
aa :=
  Panel[Row[{"ldty", {Slider[Dynamic[ldty], {100, 1000, 5}], Dynamic[ldty]}, "   rdty",
       {Slider[Dynamic[rdty], {100, 1000, 5}], Dynamic[rdty]}}],
   "Speed Settings (Sx)"];

bb := Framed[ButtonBar[{ "Sx" :> Sx[ldty, rdty], "S1" :> S1, "S2" :> S2, "S3" :> S3,
     "S4" :> S4, "FRWD" :> FRWD, "BKWD" :> BKWD, "RRWD" :> RRWD, "RLWD" :> RLWD},
```

```
    ImageMargins → 10, Background → LightGreen], Background → LightYellow];


cc := Framed[Row[{Button["START", {START, msg$ = msg2}, Background → Green],
     ButtonBar[{ " GO " :→ GO, "M1" :→ sDmd[1, ldty], "M2" :→ sDmd[2, rdty] },
      ImageMargins → 5, Background → LightOrange],
     Button["REBOOT", {R, msg$ = msg3}, Background → Red]}], Background → White];

dd := Pane[ButtonBar[
    {"Forward" :→ F, "RotateLeft" :→ RL, "STOP" :→ S, "TurnLeft" :→ TL, "BackToLeft" :→ BTL,
     "VeerLeft" :→ VL, "RampUp" :→ RU}, ImageMargins → 5, Background → LightGreen]];

ee := Pane[ButtonBar[
    {"Backward " :→ B, "RotateRight" :→ RR, " Halt " :→ H, "TurnRight" :→ TR, "BackToRight" :→ TR,
     "VeerRight" :→ VR, "RampDown" :→ RD}, ImageMargins → 5, Background → LightGreen]];

ff := Grid[{{dd}, {ee}}, Frame → All, Background → LightYellow];

gg := Panel[Row[{"dt ", {Slider[Dynamic[dt], {25, 500, 5}], Dynamic[dt]},
     " n ", {Slider[Dynamic[n], {0, 50, 1}], Dynamic[n]}}],
   "Time Increment (ms) and Step No." ];

hh := Panel[Row[{"Inch ", {Slider[Dynamic[in], {0, 50
        , 1}], Dynamic[in]}, " Deg ", {Slider[Dynamic[deg], {0, 180, 5}], Dynamic[deg]}}],
   "Inches and Degrees" ];

ii :=
  Framed[Panel[ButtonBar[{ "GODT" :→ GODT[dt], "STEP" :→ STEP[n, dt], "GF" :→ sDmd[6, in],
      "GB" :→ sDmd[7, in], "GRR" :→ sDmd[8, deg], "GRL" :→ sDmd[9, deg],
      "GTR" :→ sDmd[10, deg], "GTL" :→ sDmd[11, deg]}, ImageMargins → 5,
     Background → LightGreen, Method → "Queued"], Background → LightYellow]];

jj := Framed[
   Grid[{{Button["MAN1", {msg$ = msg4, sMan[1]}], Button["MAN2", {msg$ = msg5, sMan[2]}],
      Button["MAN3", sMan[3]]}, {Button["MAN4", sMan[4]],
      Button["MAN5", sMan[5]], Button["MAN6", sMan[6]]}}, Spacings → {0, 0}],
     Background → LightYellow];

kk := Framed[
   Panel[ Row[{Button["PING", xi = sRd[1]], InputField[Dynamic[xi], FieldSize → {4, 1}],
      "     ", "Degrees", Slider[Dynamic[pos], {-90, 90, 5}], Dynamic[pos],
      "  ", Button["SERVO", sDmd[5, pos]], "          ", Button["GamePad",
       {msg$ = msg6, nx = 1;  XBOX}, Method → "Queued"]}], Background → LightYellow]];

ll := Framed[Row[
    {Button["LENC", lenc = sRd[2]], "  ", InputField[Dynamic[lenc], FieldSize → {4, 1}],
     Button["RENC", renc = sRd[3]], " ", InputField[Dynamic[renc], FieldSize → {4, 1}],
     "  ", Button["ZENC", zenc]}], Background → White];

mm := Row[{InputField[Dynamic[msg$], Appearance → "DialogBox", FieldSize → {30, 1} ],
    "    Time (sec): ", InputField[te, FieldSize → {3, 1}] }];
```

In[65]:=
```
bx1 := ControllerState["B1"]
bx2 := ControllerState["B2"]
bx3 := ControllerState["B3"]
bx4 := ControllerState["B4"]
bx5 := ControllerState["B5"]
bx6 := ControllerState["B6"]
bx7 := ControllerState["B7"]
bx8 := ControllerState["B8"]
bx9 := ControllerState["B9"]
bx10 := ControllerState["B10"]

sx1 := ControllerState["X1"]
sy1 := ControllerState["Y1"]
sx2 := ControllerState["X2"]
sy2 := ControllerState["Y2"]
sx3 := ControllerState["X3"]
sy3 := ControllerState["Y3"]

XBOX := While[nx > 0, If[sx3 == -1, S1]; If[sx3 == 1, S3];
  If[sy3 == 1, S2]; If[sy3 == -1, S4]; If[bx1, F]; If[bx2, B]; If[bx3, RR];
  If[bx4, RL]; If[bx5, TR]; If[bx6, TL]; If[bx7, H]; If[bx8, Break[]];
  If[bx10, {ld1 = Round[sx2 * 500], rd1 = Round[sy2 * 500], sDmd[1, ld1], sDmd[2, rd1]}];
  Pause[0.2]]
```

In[82]:=
```
console := Framed[
  Panel[Column[{mm, aa, bb, Row[{cc, "      ", ll}], ff, gg, hh, Row[{ii, "   ", jj}], kk}],
   Background → LightCyan]]
```

In[83]:= **console**

Out[83]=