

1

2

3

4

A

A

voeding
voeding.SchDoc

charger
charger.SchDoc

Designator
esp32.SchDoc

B

B

VM

VM_IN

C

C

D

D

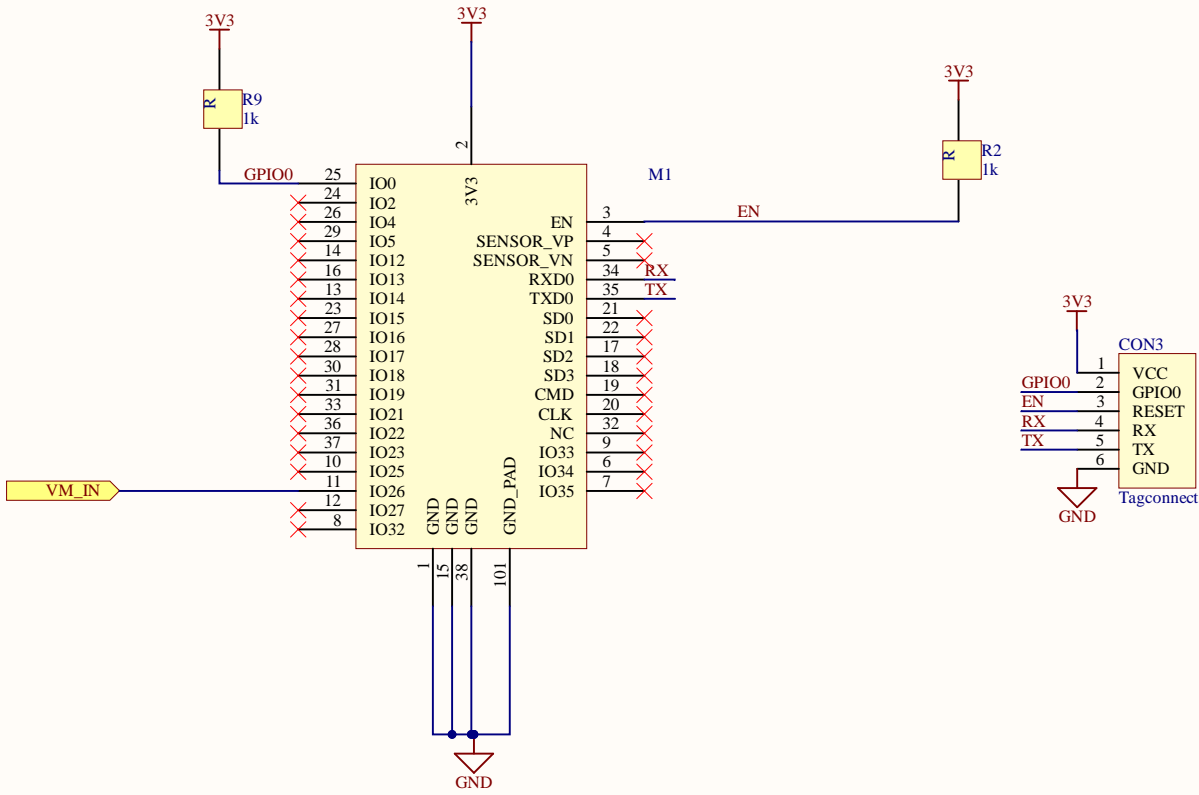
Title		
Size A4	Number	Revision
Date:	9/14/2021	Sheet of
File:	C:\Users\...\top.SchDoc	Drawn By:

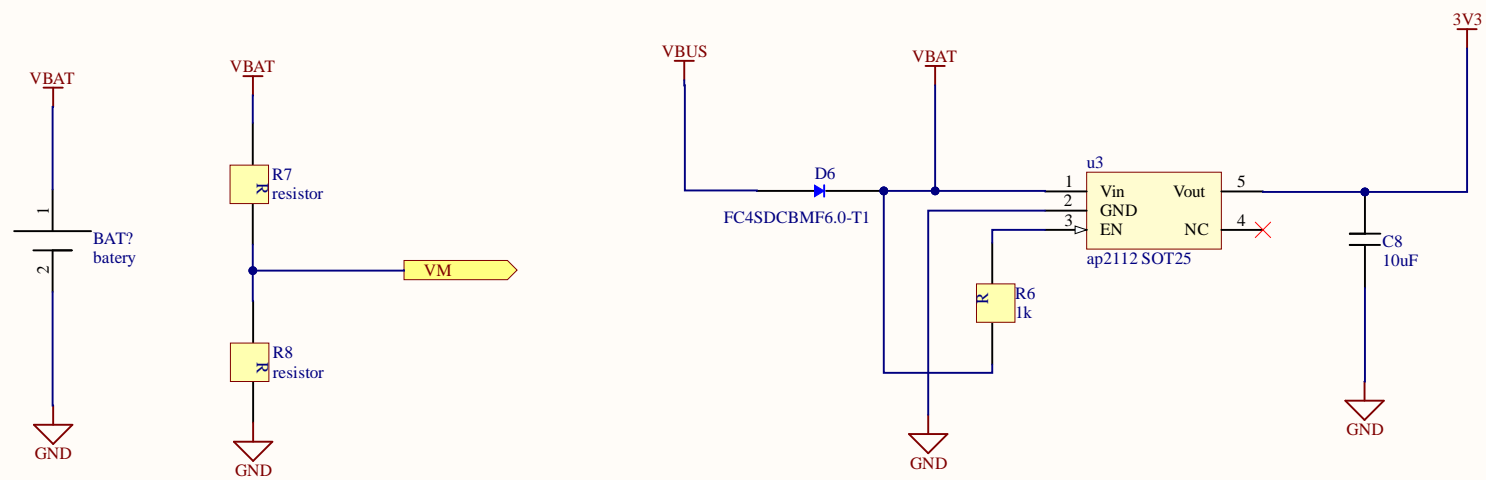
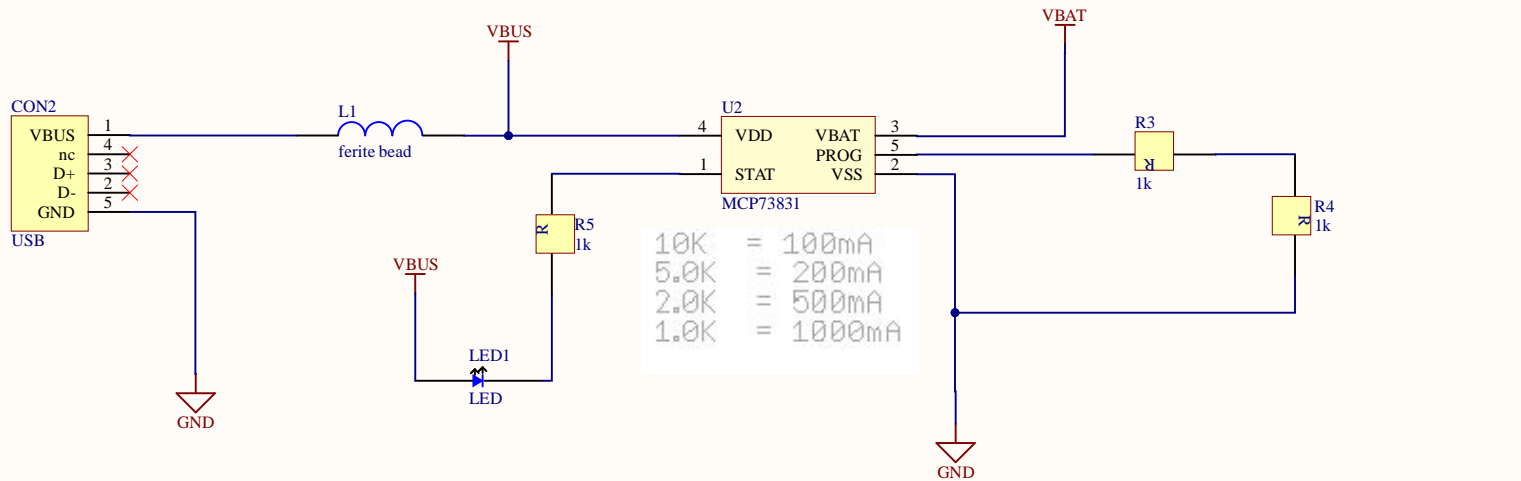
1

2

3

4

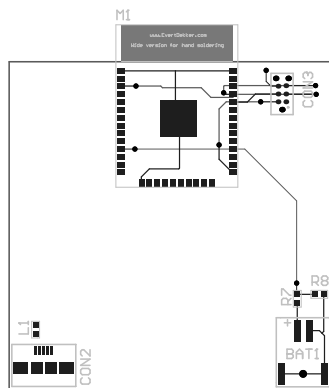




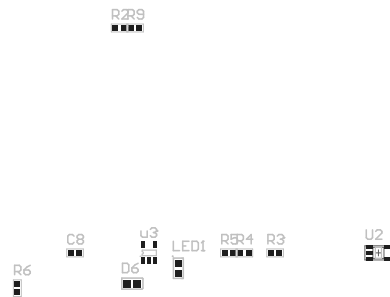
Title		
Size A4	Number	Revision
Date: 9/14/2021	Sheet of	
File: C:\Users\...\charger.SchDoc	Drawn By:	

R1 Q2 Q1 C7 C6 D5 D4 C8 C3 C4 C2 C1 D3

D2 D1 U1



6C-2L-1550um.PcbDoc
CON1



```

{ ..... }
{ Summary  This script checks checks IAR and OAR }
{          Conflicting pads and vias have been reported }
{ ..... }
{ Created by: Henk de Jonge, Transfer BV, The Netherlands }
{ Date: 15-11-2011 }
{ This script has to be used with Pattern Class 6 and Drill Class C }
{ ..... }

```

Procedure FindAnnularRingClass6C;
Var

```

Board          : IPCB_Board;
Track          : IPCB_Track;
Pad            : IPCB_Pad;
Via            : IPCB_Via;
ViaIteratorHandle : IPCB_BoardIterator;
PadIteratorHandle : IPCB_BoardIterator;
TheLayerStack  : IPCB_LayerStack;
ReportList     : TStringList;
Tolerance      : Float;
IAR            : Float;
OAR            : Float;
Via_Afm        : Float;
Hole_Afm       : Float;
Pad_AfmX       : Float;
Pad_AfmY       : Float;
Hole_Width     : Float;
AnnularRing    : Float;
Layer          : TLayer;
ViaXPos        : Float;
ViaYPos        : Float;
PadXPos        : Float;
PadYPos        : Float;
PadXAnnularRing : Float;
PadYAnnularRing : Float;
PCBOriginX     : Float;
PCBOriginY     : Float;
NumOfLayers    : Integer;

```

```

Begin
  // Obtain the PCB document interface
  Board := PCBServer.GetCurrentPCBBoard;
  If Board = Nil Then Exit;

  SetCursorBusy;

  //Showmessage('Layer: ' + FloatToStr(Layer));
  RunProcess('PCB:DeSelect');

  //Create the list with reports
  ReportList := TStringList.Create;
  ReportList.Add('Inner and Outer Annular Ring violations report:');
  ReportList.Add('_____');
  ReportList.Add('');
  ReportList.Add('VIAS');

```

```
ReportList.Add('');

// PHD = finished hole size + 0.100mm for via's (no more difference for holes <= 0.45 mm)
// IAR = 1/2 * (Inner pad diameter - PHD)
// OAR = 1/2 * (Outer pad diameter - PHD)
// ToleranceS is Tolerance for holes Smaller then 0.45mm -- not valid
// ToleranceL is Tolerance for holes Larger then 0.45mm -- not valid
```

```
Tolerance := 0.10;
OAR       := 0.125;
IAR       := 0.125;
```

```
//Obtain the PCB Current Origin, needed for X and Y coordinates
PCBOriginX := CoordToMMs(Board.XOrigin);
PCBOriginY := CoordToMMs(Board.YOrigin);
```

```
// VIAS
ViaIteratorHandle := Board.BoardIterator_Create;
ViaIteratorHandle.AddFilter_ObjectSet(MkSet(eViaObject));
ViaIteratorHandle.AddFilter_LayerSet(AllLayers);
ViaIteratorHandle.AddFilter_Method(eProcessAll);
Via := ViaIteratorHandle.FirstPCBObject;
```

```
TheLayerStack := Board.LayerStack;
NumOfLayers := TheLayerStack.SignalLayerCount;
```

```
While Via <> Nil Do
Begin
```

```
    Via.Selected := true;
    ViaXPos := CoordToMMs(Via.X) - PCBOriginX;
    ViaYPos := CoordToMMs(Via.Y) - PCBOriginY;
```

```
    // eTopLayer is 1 ; eBottomLayer is 32
    For Layer := eTopLayer to eBottomLayer Do
    Begin
```

```
        If (Layer = 1) or (Layer = 32) Then
        Begin
```

```
            Hole_Afm := CoordToMMs(Via.Holesize);
            Via_Afm := CoordToMMs(Via.SizeOnLayer[Layer]);
            If (Via_Afm <> 0) Then
            Begin
```

```
                AnnularRing := (Via_afm - Tolerance - Hole_Afm)/2;
                If (AnnularRing < OAR) Then
                Begin
                    ReportList.Add('OAR Violation for Via on ' +
                        Layer2String(Layer) +
                        ' on location: ' +
                        FloatToStr(ViaXPos) +
                        ',' +
                        FloatToStr(ViaYPos) +
                        ' ; ' +
                        'Outer Annular Ring ' +
                        FloatToStr(AnnularRing) +
                        ' is smaller dan OAR Rule: ' +
                        FloatToStr(OAR));
                End;
```

```
            End;
        End
```

```

Else
Begin
  If Layer < NumOfLayers Then
  Begin
    Hole_Afm := CoordToMMs(Via.Holesize);
    Via_Afm := CoordToMMs(Via.SizeOnLayer[Layer]);
    If (Via_Afm <> 0) Then
    Begin
      AnnularRing := (Via_afm - Tolerance - Hole_Afm)/2;
      If (AnnularRing < IAR) Then
      Begin
        ReportList.Add('IAR Violation for Via on ' +
          Layer2String(Layer) +
          ' on location: ' +
          FloatToStr(ViaXPos) +
          ',' +
          FloatToStr(ViaYPos) +
          ' ; ' +
          'Inner Annular Ring ' +
          FloatToStr(AnnularRing) +
          ' is smaller dan IAR Rule: ' +
          FloatToStr(IAR));
      End;
    End;
  End;
End;

Via.Selected := false;
Via := ViaIteratorHandle.NextPCBObject;
End;
Board.BoardIterator_Destroy(ViaIteratorHandle);

ReportList.Add('');
ReportList.Add('PADS:');
ReportList.Add('');

// PADS
PadIteratorHandle := Board.BoardIterator_Create;
PadIteratorHandle.AddFilter_ObjectSet(MkSet(ePadObject));
PadIteratorHandle.AddFilter_LayerSet(AllLayers);
PadIteratorHandle.AddFilter_Method(eProcessAll);
Pad := PadIteratorHandle.FirstPCBObject;

While Pad <> Nil Do
Begin
  Pad.Selected := true;
  PadXPos := CoordToMMs(Pad.X) - PCBOriginX;
  PadYPos := CoordToMMs(Pad.Y) - PCBOriginY;

  Hole_Width := 0.0;
  For Layer := MinLayer to MaxLayer Do
  Begin
    if (Layer = 74) Then
    Begin
      Hole_Width := CoordToMMs(Pad.HoleSize);
    End;
  End;
End;

```



```

if Hole_Width > 0 Then
Begin
    // eTopLayer is 1 ; eBottomLayer is 32
    For Layer := eTopLayer to eBottomLayer Do
    Begin
        If (Layer = 1) or (Layer = 32) Then
        Begin
            Pad_AfmX := CoordToMMs(Pad.XSizeOnLayer[Layer]);
            Pad_AfmY := CoordToMMs(Pad.YSizeOnLayer[Layer]);
            If (Pad_AfmX <> 0) And (Pad_AfmY <> 0) Then
            Begin
                PadXAnnularRing := (Pad_afmX - Tolerance - Hole_Width)/2;
                If (PadXAnnularRing < OAR) Then
                Begin
                    ReportList.Add('OAR Violation for Pad-X on ' +
                        Layer2String(Layer) +
                        ' on location: ' +
                        FloatToStr(PadXPos) +
                        ',' +
                        FloatToStr(PadYPos) +
                        ' ; ' +
                        'Outer Annular Ring ' +
                        FloatToStr(PadXAnnularRing) +
                        ' is smaller dan OAR Rule: ' +
                        FloatToStr(OAR));
                End;
            End;
            PadYAnnularRing := (Pad_afmY - Tolerance - Hole_Width)/2;
            If (PadYAnnularRing < OAR) Then
            Begin
                ReportList.Add('OAR Violation for Pad-Y on ' +
                    Layer2String(Layer) +
                    ' on location: ' +
                    FloatToStr(PadXPos) +
                    ',' +
                    FloatToStr(PadYPos) +
                    ' ; ' +
                    'Outer Annular Ring ' +
                    FloatToStr(PadYAnnularRing) +
                    ' is smaller dan OAR Rule: ' +
                    FloatToStr(OAR));
            End;
        End;
    End
Else
Begin
    If Layer < NumOfLayers Then
    Begin
        Pad_AfmX := CoordToMMs(Pad.XSizeOnLayer[Layer]);
        Pad_AfmY := CoordToMMs(Pad.YSizeOnLayer[Layer]);
        If (Pad_AfmX <> 0) And (Pad_AfmY <> 0) Then
        Begin
            PadXAnnularRing := (Pad_afmX - Tolerance - Hole_Width)/2;
            If (PadXAnnularRing < IAR) Then
            Begin
                ReportList.Add('IAR Violation for Pad-X on ' +
                    Layer2String(Layer) +
                    ' on location: ' +
                    FloatToStr(PadXPos) +

```

```

        ', ' +
        FloatToStr(PadYPos) +
        ' ; ' +
        'Inner Annular Ring ' +
        FloatToStr(PadXAnnularRing) +
        ' is smaller dan IAR Rule: ' +
        FloatToStr(IAR));
    End;
    PadYAnnularRing := (Pad_afmY - Tolerance - Hole_Width)/2;
    If (PadYAnnularRing < IAR) Then
    Begin
        ReportList.Add('IAR Violation for Pad-Y on ' +
            Layer2String(Layer) +
            ' on location: ' +
            FloatToStr(PadXPos) +
            ', ' +
            FloatToStr(PadYPos) +
            ' ; ' +
            'Inner Annular Ring ' +
            FloatToStr(PadYAnnularRing) +
            ' is smaller dan IAR Rule: ' +
            FloatToStr(IAR));
    End;
    End;
    End;
    End;
    End;
    Pad.Selected := false;
    Pad := PadIteratorHandle.NextPCBObject;

End;
ReportList.SaveToFile('C:\TEMP\AnnularRing.txt');
ReportList.Free;
Showmessage('Output completed. See C:\TEMP\AnnularRing.txt for results.');
```

```

Board.BoardIterator_Destroy(PadIteratorHandle);
ResetCursor;
```

```

End;
{.....}
```

Comment	Description	Designator	Footprint	LibRef	Quantity
battery		BAT?	jst 2pin	battery	1
0,022uF		C1, C5	0603	capacitor	2
470pF		C2	0603	capacitor	1
10uF		C3, C4, C8	0603	capacitor	3
47nF		C6, C7	CAPC1005X04L	capacitor	2
screw connector	screw connector	CON1	qi coil	screw connector	1
USB		CON2	microusb	USB	1
Tagconnect		CON3	Tagconnect	Tagconnect	1
MBR0540-TP		D1, D2, D3, D4, D5	Diode	diode	5
FC4SDCBMF6.0-T1		D6	Diode	diode	1
ferite bead		L1	0603	Spoel	1
LED		LED1	LED	LED	1
ESP32S_Module	Esp-32S	M1	ESP32S_W	ESP32S_Module	1
2N7002ET1G		Q1, Q2	N-channel FET	fet	2
10K		R1	CAPC1005X04L	resistor	1
1k		R2, R3, R4, R5, R6, R9	0603	resistor	6
resistor		R7, R8	0603	resistor	2
ap2112 SOT25		u3	AP2112 sot25	ap2112 SOT25	1
MCP1755T-500E/OT		U1	AP2112 sot25	MCP1755T-500E/OT	1
MCP73831		U2	MCP73831/2	MCP73831	1