

```

package net.minecraft.src;

import java.io.*;

// Created by MoareAI

public class mod_DigitalFunctions extends BaseMod
{
    //ModLoader actions
    public mod_DigitalFunctions()
    {
        RenderFunctionID = ModLoader.getUniqueBlockModelID(this, false);
        ModLoader.RegisterTileEntity(MoareAI.Class.TileEntityDigitalMisc.class,
        "DigitalFunctions");

        //Item names
        ModLoader.AddName(ItemDigitalMisc, "Digital Functions");
        for(int damage = 0; damage < 9; damage++)
        {
            ModLoader.AddName(new ItemStack(ItemDigitalMisc, 1, damage),
            ItemDigitalMisc.getItemNameIS(new ItemStack(ItemDigitalMisc, 1, damage)));
        }

        if (EffectiveCrafting)
            CraftCost = 1;
        else
            CraftCost = 0;

        //Recipe for Toggle
        ModLoader.AddRecipe(new ItemStack(ItemDigitalMisc, CraftCost*5+1, 1), new
Object[] {
            " Z ", "#XY", Character.valueOf('#'), Item.redstone,
            Character.valueOf('X'), Block.sand, Character.valueOf('Y'), Block.torchRedstoneActive,
            Character.valueOf('Z'), Block.lever
        });

        //Recipe for RS
        ModLoader.AddRecipe(new ItemStack(ItemDigitalMisc, CraftCost*2+1, 2), new
Object[] {
            "XY#", "# #", "#YX", Character.valueOf('#'), Item.redstone,
            Character.valueOf('X'), Block.sand, Character.valueOf('Y'), Block.torchRedstoneActive
        });

        //Recipe for Pulse Clock
        ModLoader.AddRecipe(new ItemStack(ItemDigitalMisc, CraftCost*5+1, 3), new
Object[] {
            "#XY", Character.valueOf('#'), Item.pocketSundial, Character.valueOf('X'),
            Block.sand, Character.valueOf('Y'), Block.torchRedstoneActive
        });

        //Recipe for JK Flip Flop
        ModLoader.AddRecipe(new ItemStack(ItemDigitalMisc, 1, 5), new Object[] {
            "#X", Character.valueOf('#'), new ItemStack(ItemDigitalMisc, 1, 2),
            Character.valueOf('X'), new ItemStack(ItemDigitalMisc, 1, 1)
        });

        //Recipe for Pulse Generator
        ModLoader.AddRecipe(new ItemStack(ItemDigitalMisc, 1, 6), new Object[] {
            " #", "X ", Character.valueOf('#'), new ItemStack(ItemDigitalMisc, 1, 2),
            Character.valueOf('X'), Item.redstoneRepeater
        });

        //Recipe for Counter
        ModLoader.AddRecipe(new ItemStack(ItemDigitalMisc, 1, 7), new Object[] {

```

mod_DigitalFunctions.java

```
        "#X", Character.valueOf('#'), new ItemStack(ItemDigitalMisc, 1, 4),
        Character.valueOf('X'), new ItemStack(ItemDigitalMisc, 1, 1)
    });

    //Random Generator
    ModLoader.AddRecipe(new ItemStack(ItemDigitalMisc, CraftCost*2+1, 8), new
Object[] {
        "#XY", Character.valueOf('#'), Block.leaves, Character.valueOf('X'),
        Block.sand, Character.valueOf('Y'), Block.torchRedstoneActive, Character.valueOf('Z'),
        Block.lever
    });
}

public void ModsLoaded()
{
    if (ModLoader.isModLoaded("mod_LogicalGates"))
    {
        //Recipe for Half Adder
        ModLoader.AddRecipe(new ItemStack(ItemDigitalMisc, 1, 4), new Object[] {
            "#", "X", Character.valueOf('#'), new
ItemStack(mod_LogicalGates.ItemLogicalGates, 1, 5), Character.valueOf('X'), new
ItemStack(mod_LogicalGates.ItemLogicalGates, 1, 4)
        });

        //Recipe for Pulse Generator
        ModLoader.AddRecipe(new ItemStack(ItemDigitalMisc, 1, 6), new Object[] {
            " #", "YX ", Character.valueOf('#'), new
ItemStack(mod_LogicalGates.ItemLogicalGates, 1, 4), Character.valueOf('X'), new
ItemStack(mod_LogicalGates.ItemLogicalGates, 1, 2), Character.valueOf('Y'),
Item.redstoneRepeater
        });
    }
}

public String Version()
{
    return "3.1.2";
}

//Block declaration
public static final Block BlockDigitalMisc;

//Render type declaration
public static int RenderFunctionID;

//Item declaration
public static Item ItemDigitalMisc;

//Properties
public static final File cfgfile;
public static int PulseGenLength = 2;
public static int ClockTime = 16;
public static int DefaultCount = 5;
public static boolean EffectiveCrafting = true;
private int CraftCost;
public static boolean ReduceIcon = false;
public static int GUIStep = 5;
public static int DigitalFunctionsBlockID = 201;
public static int DigitalFunctionsItemID = 457;
public static String BlockTexture = "/MoareAI/Blocks/DigitalFunctions.png";

//Block and Item data
static
{

```

mod_DigitalFunctions.java

```
//The configuration file
cfgfile = new File(Minecraft.getMinecraftDir(), "/config/MoareAI's Digital
Functions.cfg");
ConfigDigitalFunctions.Configure(cfgfile);

BlockDigitalMisc = new
BlockDigitalMisc(DigitalFunctionsBlockID).setBlockName("MiscDigital");
ItemDigitalMisc = new ItemDigitalMisc(DigitalFunctionsItemID,
BlockDigitalMisc).setItemName("MiscDigital");
}

//The rendering for the gates
public boolean RenderWorldBlock(RenderBlocks renderblocks, IBlockAccess
iblockaccess, int x, int y, int z, Block block, int l)
{
    Tessellator tessellator = Tessellator.instance;

    if(l == RenderFunctionID)
    {
        byte A = MoareAI.Class.BlockDigitalMisc.A;
        byte B = MoareAI.Class.BlockDigitalMisc.B;
        byte C = MoareAI.Class.BlockDigitalMisc.C;
        byte F = MoareAI.Class.BlockDigitalMisc.F;
        byte G = MoareAI.Class.BlockDigitalMisc.G;
        byte I = MoareAI.Class.BlockDigitalMisc.I;
        TileEntityDigitalMisc tileentity =
(TileEntityDigitalMisc)iblockaccess.getBlockTileEntity(x, y, z);
        int meta = iblockaccess.getBlockMetadata(x, y, z);
        int rotation = tileentity.Rotation;
        float light = block.getBlockBrightness(iblockaccess, x, y, z);
        float white = 1.0F;
        //Gate Model
        RenderFunction.RenderFunctionModel(tessellator, block, iblockaccess, x, y,
z, white, light, rotation);

        if (meta > 0)
        {
            float red = 0.5F;

            //Output F
            red = 0.5F;
            if (tileentity.Output[F])
                red = 1.0F;
            RenderFunction.RenderOutput(tessellator, block, x, y, z, red, light,
rotation, 0);
            RenderFunction.RenderConnectionPiece(tessellator, block, x, y, z, red,
light, rotation, 0);

            //Output G
            if (((meta == 2 || meta == 4) && !tileentity.State[2]) || meta == 8)
            {
                red = 0.5F;
                if (tileentity.Output[G])
                    red = 1.0F;
                RenderFunction.RenderOutput(tessellator, block, x, y, z, red,
light, rotation, 1);
                RenderFunction.RenderConnectionPiece(tessellator, block, x, y, z,
red, light, rotation, 1);
            }

            //Output I
            if (meta == 8 || ((meta == 2 || meta == 4) && tileentity.State[2]))
            {
                red = 0.5F;
```

```

mod_DigitalFunctions.java

    if (tileentity.Output[I])
        red = 1.0F;
    RenderFunction.RenderOutput(tessellator, block, x, y, z, red,
light, rotation, I);
    RenderFunction.RenderConnectionPiece(tessellator, block, x, y, z,
red, light, rotation, I);
}

//Input Connection
    if (tileentity.Output[F] || tileentity.Output[G] ||
tileentity.Output[I])
        red = 1.0F;
    RenderFunction.RenderInputConnection(tessellator, block, x, y, z, red,
light, rotation);

//InputA
    float cyan = 0.0F;
    if (meta == 5 || meta == 7 || ((meta == 2 || meta == 4) &&
tileentity.State[2]))
    {
        red = 0.5F;
        if (tileentity.Input[A])
        {
            red = 1.0F;
        }
        if (!tileentity.Connection[A])
        {
            cyan = 0.5F;
        }
    }
    RenderFunction.RenderInput(tessellator, block, x, y, z, red, cyan,
light, rotation, 3);
}

//InputB
    cyan = 0.0F;
    red = 0.5F;
    if (tileentity.Input[B])
    {
        red = 1.0F;
    }
    if (!tileentity.Connection[B])
    {
        cyan = 0.5F;
    }
    RenderFunction.RenderInput(tessellator, block, x, y, z, red, cyan,
light, rotation, 0);

//InputC
    if (((meta == 2 || meta == 4) && !tileentity.State[2]) || meta == 5 ||
meta == 7)
    {
        cyan = 0.0F;
        red = 0.5F;
        if (tileentity.Input[C])
        {
            red = 1.0F;
        }
        if (!tileentity.Connection[C])
        {
            cyan = 0.5F;
        }
    }
    RenderFunction.RenderInput(tessellator, block, x, y, z, red, cyan,
light, rotation, 1);

```

mod_DigitalFunctions.java

```
        }  
    }  
}  
//if(block instanceof ICustomTextureBlock)  
//{  
//    Tessellator tessellator1 = Tessellator.instance;  
//    tessellator1.draw();  
//    tessellator1.startDrawingQuads();  
//    GL11.glBindTexture(3553 /*GL_TEXTURE_2D*/,  
ModLoader.getMinecraftInstance().renderEngine.getTexture("/terrain.png"));  
//    return true;  
//}  
return false;  
}
```