int APIENTRY WinMain(HINSTANCE hInstance,

HINSTANCE hPrevInstance,

LPTSTR lpCmdLine,

int nCmdShow)

{

Adapter::rpmEntries= cfg.GetTokens<std::string>("GLOBALS.RPMTAGS", ",");

Adapter::bAllFakeSpindle = cfg.GetSymbolValue<int>("GLOBALS.AllFakeSpindle", "0");

void Adapter::CheckAlias(Device \*device, std::string &key, std::string &value)

{

////////////////////////////////////////////////////////////////////////////////

// Michaloski hack - if srpm2 is > 0 assign it to Srpm

if(std::find(rpmEntries.begin(), rpmEntries.end(), key)!= rpmEntries.end())

{

try {

int rpm;

try {

rpm = toNumber<int>(value);

}

catch(...){ rpm=0; }

if(rpm>0)

key="Srpm";

}

catch(...)

{

// bummer conversion didn't work

}

}

// Michaloski hack - fake a spindle in case there is none

// This should be in a separate thread but we'll live

// time was overloaded replaced with shdrtime

if((bAllFakeSpindle==1 || bDeviceFakeSpindle[device->getName()] == 1) && (difftime( time(0), lastime) > 1.0) )

{

// LOG\_ONCE(logFatal("Fake Spindle Enabled\n") );

// Reset second calculator

lastime=time(0);

try {

// First we sample xyz mode/execution feed as often as we can.

// We assume xyz will change frequently enough

DataItem \*sRpmDataItem = device->getDeviceDataItem("Srpm");

if(sRpmDataItem==NULL)

throw(std::string("Null Srpm DeviceDataItem"));

std::string feed,X, Y, Z, mode,execution;

DataItem \* dExecution = device->getDeviceDataItem("execution");

DataItem \* dMode = device->getDeviceDataItem("controllermode");

DataItem \* dXabs = device->getDeviceDataItem("Xabs");

DataItem \* dYabs = device->getDeviceDataItem("Yabs");

DataItem \* dZabs = device->getDeviceDataItem("Zabs");

DataItem \* dFeed = device->getDeviceDataItem("path\_feedratefrt");

if(dExecution==NULL || dMode==NULL || dFeed==NULL || dXabs==NULL ||

dXabs==NULL || dYabs==NULL || dZabs==NULL )

throw(std::string("Null Fake RPM DeviceDataItem"));

std::set<string> aFilter ;

aFilter.insert(dExecution->getId());

aFilter.insert(dMode->getId());

aFilter.insert(dXabs->getId());

aFilter.insert(dYabs->getId());

aFilter.insert(dZabs->getId());

aFilter.insert(dFeed->getId());

ComponentEventPtrArray events;

mAgent->mLatest.getComponentEvents(events, &aFilter);

// Dubious error check

//if(events.size() < 6)

// throw(std::string("Bad Number of Fake ComponentEventsPtr"));

execution = events[0] ->getValue();

mode = events[1] ->getValue();

X = events[2] ->getValue();

Y = events[3] ->getValue();

Z = events[4] ->getValue();

feed = events[5] ->getValue();

// If moving and in auto mode, assume spindle on

if( (mode == "AUTOMATIC" ) &&

(execution == "EXECUTING" )&&

(

device->lastFeed!=feed ||

device->lastX!=X ||

device->lastY!= Y ||

device->lastZ!=Z

)

)

{

// Fixme: this should be some time based calculation

device->mLag=4; // 3 cycles - 3 seconds

}

else

{

device->mLag--;

}

if(device->mLag<0)

device->mLag=0;

if(device->mLag>0)

{

// set spindle speed to fake number

mAgent->addToBuffer(sRpmDataItem, "99", shdrtime);

}

else

{

// set spindle speed to zero

mAgent->addToBuffer(sRpmDataItem, "0", shdrtime);

}

}

catch(std::string & errmsg)

{

// logFatal(errmsg.c\_str());

}

catch(...)

{

}

}