Forced definition of struct timezone under MinGW

It appears that win32.h is forcing the definition of timezone structure, even when it has been defined by MinGW provided headers.

The output from compilation of miniruby:

```
In file included from ../include/ruby/defines.h:205:0,
    from ../include/ruby/ruby.h:73,
    from ../include/ruby.h:32,
    from ../main.c:13:
../include/ruby/win32.h:201:8: error: redefinition of 'struct timezone'
And the code that defines it:

#define MINGW32
struct timezone {
    int tz_minuteswest;
    int tz_dsttime;
};
#undef isascii
#define isascii __isascii
#endif

It is clear that this was made to workaround limitations of older versions of MinGW, but a simple check with AC_CHECK_TYPES (or if specific are needed: AC_CHECK_MEMBERS), any of those could provide the defines needed to avoid the forced redefinition.

Please find attached a naive patch for configure.in and win32.h that use HAVE_STRUCT_TIMEZONE

Thank you.
```

Related issues:

Related to Backport187 - Backport #3666: Backport of r26311 (Bug #2587)

History

#1 - 01/12/2010 07:48 AM - rogerdpack (Roger Pack)

```
begin
is this a cross compile? Which version of gcc?
end
```

#2 - 01/12/2010 07:54 AM - luislavena (Luis Lavena)

```
begin
Roger:

No cross-compiler, native.

Timezone is part of the newer w32api packages for MinGW. Will grab the versions and follow once I have the computer around.
```

09/17/2021
This issue was solved with changeset r26311. Luis, thank you for reporting this issue. Your contribution to Ruby is greatly appreciated. May Ruby be with you.

Files

| check-timezone-struct-avoid-redefinition.patch | 956 Bytes | 01/10/2010 | luislavena (Luis Lavena) |