Make Your Windows 11 PC Speak the Time Every Hour (Male/Female Voice) — Step-by-Step Guide

If you want your Windows 11 computer to announce the time automatically at every hour, this guide will help you set it up in a clean, reliable way — without installing any third-party software.

We will use:

PowerShell (built into Windows)

Task Scheduler (built into Windows)

Windows Text-to-Speech voice (male or female, depending on what's installed)

By the end, your PC will speak:

"The time is 03 00 PM"
…at every hour, as long as the computer is ON.

Part A — Create the "Speak Time" Script (PowerShell)

Step 1: Create a folder for your script

Open File Explorer

Go to *C:*

Create a new folder named:

Scripts

☐ **[Screenshot 1: File Explorer showing C:\Scripts folder created]**

Step 2: Create the PowerShell script file

1. Open **Notepad**
2. Paste the following code:

```
Add-Type -AssemblyName System.Speech
$speaker = New-Object System.Speech.Synthesis.SpeechSynthesizer
$time = Get-Date -Format "hh mm tt"
$speaker.Speak("The time is $time")
```

3. Click **File → Save As**
4. Save it as:
5. `speak_time.ps1`
6. Save location:
7. `C:\Scripts\`
8. In "Save as type", choose **All Files**
9. Encoding can remain **UTF-8**

☐ **[Screenshot 2: Notepad "Save As" showing speak_time.ps1 and Save as type = All Files]**

☑At this point you have:

`C:\Scripts\speak_time.ps1`

Part B — Select Male/Female Voice (Optional, but Recommended)

Windows uses installed Text-to-Speech voices. Many PCs will have only one voice by default, but you can install more voices.

Step 3: Check and install voices in Windows 11

1. Open **Settings**
2. Go to:
   Accessibility → Speech
3. Look for **Voices**
4. If you see multiple voices, great.
5. If you want more voices, use:
   Add voices (if available)

☐ **[Screenshot 3: Settings → Accessibility → Speech → Voices section]**

Step 4: Modify the script to force a specific voice (Male/Female)

If your PC has multiple voices installed, you can choose which one to use.

Open `C:\Scripts\speak_time.ps1` in Notepad and use this version:

```
Add-Type -AssemblyName System.Speech
$speaker = New-Object System.Speech.Synthesis.SpeechSynthesizer

# OPTIONAL: choose a specific voice (male/female depends on installed voices)
# To see available voices, run this in PowerShell:
```

```
# (New-Object
System.Speech.Synthesis.SpeechSynthesizer).GetInstalledVoices() |
ForEach-Object { $_.VoiceInfo.Name }

# Example voice selection (replace with your installed voice name)
# $speaker.SelectVoice("Microsoft Zira Desktop")

$time = Get-Date -Format "hh mm tt"
$speaker.Speak("The time is $time")
```

How to pick the correct voice name:

1. Open **PowerShell**
2. Paste and run:

```
(New-Object
System.Speech.Synthesis.SpeechSynthesizer).GetInstalledVoices() |
ForEach-Object { $_.VoiceInfo.Name }
```

This prints the exact voice names installed on your PC. Copy one and place it inside:

```
$speaker.SelectVoice("VOICE NAME HERE")
```

☐ **[Screenshot 4: PowerShell window showing the list of installed voices]**

☐ **[Screenshot 5: Notepad showing SelectVoice line with chosen voice name]**

Note: Voice gender depends on what Microsoft provides and what your system language pack includes.

Part C — Create the Hourly Task in Task Scheduler

Now we'll tell Windows: "Run this script every hour."

Step 5: Open Task Scheduler

1. Press **Win + R**
2. Type:
3. taskschd.msc
4. Press **Enter**

☐ **[Screenshot 6: Run dialog with taskschd.msc]**

Step 6: Create a new task (Basic Task)

1. In the right panel, click **Create Basic Task…**
2. Name it:
3. Hourly Time Announcer
4. Optionally add description:
5. Speaks the current time every hour while the PC is on.

☐ **[Screenshot 7: Create Basic Task → Name and Description]**

Step 7: Set the trigger (repeat every hour)

1. Select **Daily**
2. Click **Next**
3. Set any start date/time (example: today, 09:00 AM)
4. Click **Next**
5. On the "Daily" schedule screen:
   - Tick: **Repeat task every:** 1 hour
   - Duration: **Indefinitely**

☐ **[Screenshot 8: Trigger settings showing repeat every 1 hour indefinitely]**

☑This ensures it runs every hour continuously.

Step 8: Set the action (run PowerShell)

1. Choose: **Start a program**
2. Program/script:
3. powershell.exe
4. Add arguments (IMPORTANT — adjust path if needed):
5. -ExecutionPolicy Bypass -File "C:\Scripts\speak_time.ps1"
6. Start in:
7. C:\Scripts

☐ **[Screenshot 9: Action screen showing powershell.exe + arguments + Start in]**

Step 9: Finish

Click **Finish**.

Now the task is created — but we'll do two quick improvements for reliability.

Part D — Improve Reliability (Recommended Settings)

Step 10: Open task properties

1. In Task Scheduler, click **Task Scheduler Library**
2. Find **Hourly Time Announcer**
3. Right-click → **Properties**

☐ **[Screenshot 10: Task Scheduler Library with the created task selected]**

Step 11: Configure these important options

In the General tab:

- ☑**Run whether user is logged on or not**
- ☑**Run with highest privileges**

☐ **[Screenshot 11: Task Properties → General tab showing these options]**

In the Conditions tab (Laptop users):

- If you want it to work on battery too, uncheck:
  - ✖"Start the task only if the computer is on AC power"

☐ **[Screenshot 12: Task Properties → Conditions tab (AC power checkbox)]**

Click **OK** to save.

Part E — Test It Now

Step 12: Run the task manually (test)

1. Go to **Task Scheduler Library**
2. Right-click **Hourly Time Announcer**
3. Click **Run**

You should immediately hear the computer speak the time.

☐ **[Screenshot 13: Right click menu → Run]**

Customization Options (Optional)

1) Change the spoken sentence

Replace this line:

```
$speaker.Speak("The time is $time")
```

with something like:

```
$speaker.Speak("Attention. The time is now $time")
```

2) Use 24-hour format

Replace:

```
$time = Get-Date -Format "hh mm tt"
```

with:

```
$time = Get-Date -Format "HH mm"
```

3) Make it speak only during a time range (advanced idea)

If you want announcements only between (say) **8 AM to 10 PM**, you can add a simple hour-check in script.

Example:

```
$h = (Get-Date).Hour
if ($h -ge 8 -and $h -le 22) {
    $speaker.Speak("The time is $time")
}
```

Troubleshooting (Common Issues)

"Nothing happens when I run it"

Check these:

- Is the script path correct?
    - C:\Scripts\speak_time.ps1
- Are the PowerShell arguments exactly correct?

- o `-ExecutionPolicy Bypass -File "C:\Scripts\speak_time.ps1"`
- Test the script directly:
  1. Open PowerShell
  2. Run:
  3. `powershell -ExecutionPolicy Bypass -File "C:\Scripts\speak_time.ps1"`

☐ **[Screenshot 14: PowerShell test command running successfully]**

"It runs but voice is not what I want"

- Install another voice in **Settings → Accessibility → Speech**
- List installed voices via PowerShell
- Use `SelectVoice("...")` with exact voice name

Conclusion

You've now configured your Windows 11 PC to:

- Speak time automatically
- Run every hour (while the PC is ON)
- Use male/female voice depending on installed voices
- Work reliably via Windows Task Scheduler